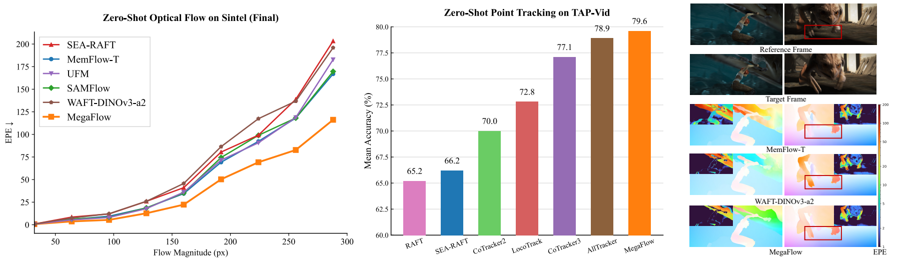
MegaFlow: Zero-Shot Large Displacement Optical Flow

Dingxi Zhang[1], Fangjinhua Wang[1], Marc Pollefeys[1,2], and Haofei Xu[1]

[1]ETH Zurich [2]Microsoft



(a) Zero-shot optical flow (b) Zero-shot point tracking (c) Large displacement flow

Fig. 1: MegaFlow excels at large displacement optical flow and point tracking. (**a**) On the Sintel (Final) benchmark, MegaFlow consistently achieves the lowest End-Point Error (EPE), with its advantage widening significantly on large displacements. (**b**) MegaFlow also demonstrates superior zero-shot point tracking results on TAP-Vid. (**c**) Visuals and inset error maps further illustrate our state-of-the-art results.

Abstract. Accurate estimation of large displacement optical flow remains a critical challenge. Existing methods typically rely on iterative local search or/and domain-specific fine-tuning, which severely limits their performance in large displacement and zero-shot generalization scenarios. To overcome this, we introduce MegaFlow, a simple yet powerful model for zero-shot large displacement optical flow. Rather than relying on highly complex, task-specific architectural designs, MegaFlow adapts powerful pre-trained vision priors to produce temporally consistent motion fields. In particular, we formulate flow estimation as a global matching problem by leveraging pre-trained global Vision Transformer features, which naturally captures large displacements. This is followed by a few lightweight iterative refinement to further improve the sub-pixel accuracy. Extensive experiments demonstrate that MegaFlow achieves state-of-the-art zero-shot performance across multiple optical flow benchmarks. Moreover, our model also delivers highly competitive zero-shot performance on long-range point tracking benchmarks, demonstrating its robust transferability and suggesting a unified paradigm for generalizable motion estimation. Project Page: `https://kristen-z.github.io/projects/megaflow/`.

Keywords: Optical Flow · Large Displacement · Point Tracking

1 Introduction

Optical flow estimation is a fundamental problem in computer vision, providing dense pixel-level correspondences that are essential for a wide range of applications, such as autonomous driving [12, 34] and 3D reconstruction [31, 59].

Traditional methods typically formulate optical flow as an optimization problem [16, 29, 49], but struggle with large motions and complex appearance variations. Deep learning dramatically advances this field: early CNN-based approaches [11, 40] demonstrate end-to-end optical flow prediction, while PWC-Net [50] incorporates pyramid warping and cost volumes to handle large displacements. RAFT [52] further establishes a powerful paradigm using local iterative refinements, inspiring a family of variants emphasizing improved feature aggregation and efficiency [18, 57, 58]. In parallel, global correspondence matching emerges to complement local refinement [53, 63, 64, 67], and Transformer-based architectures demonstrate the potential of long-range feature representations for accurate and robust optical flow estimation [17, 47].

Over the years, the field has advanced considerably, driven by dedicated datasets [23, 33, 34] and increasingly sophisticated model designs [17, 46, 51]. Despite these advances, existing methods face two distinct bottlenecks: (1) reliance on task-specific features and domain-specific fine-tuning, which limits out-of-distribution generalization. (2) an architectural vulnerability in resolving large displacements, as iterative local search suffers from severe ambiguities across massive spatial gaps. Consequently, developing a highly accurate and universally generalizable foundation model for optical flow remains an open challenge.

In this paper, we propose MegaFlow, a general optical flow framework designed for large displacement motion estimation and strong zero-shot generalization. MegaFlow is inspired by large vision architectures [22, 54, 56], whose alternating frame-wise and global-attention Transformer blocks have shown highly effective at modeling cross-view relations. To extend these geometric priors trained on static scenes to dynamic motion estimation, we design a simple yet powerful architecture that builds a globally consistent representation for multi-frame motion estimation. Crucially, this unified design is highly adaptable to diverse vision foundation models and generalizes effectively to different tasks. Instead of directly regressing flow vectors [25, 59, 66], we first perform global matching to establish accurate initial correspondences. These globally informed predictions are then injected into a lightweight iterative refinement module that uses local correlation to further improve accuracy while preserving fine-grained details. MegaFlow achieves state-of-the-art zero-shot performance on various optical flow and point tracking benchmarks, effectively bridging the gap between powerful static geometric priors and dynamic, large displacement motion estimation.

Our main contributions are summarized as follows:

1. We introduce MegaFlow, a simple yet powerful framework for optical flow that significantly improves large displacements and zero-shot generalization.
2. We effectively adapt static pre-trained vision priors to dynamic motion estimation. Our architecture seamlessly integrates global matching with lightweight iterative refinement to capture both large displacements and fine-grained sub-pixel details.
3. Extensive experiments show that MegaFlow sets a new state of the art for zero-shot optical flow and demonstrates strong cross-task generalization through robust zero-shot point tracking, while also delivering robust performance on in-the-wild video sequences.

2 Related Work

Optical Flow Estimation. Classical methods typically formulate optical flow as an energy minimization problem [16, 29], but they usually struggle with complex appearance variations. Early deep learning models enable end-to-end prediction via CNNs and pyramid-based cost volumes [11, 40, 50]. Subsequently, RAFT [52] establishes the dominant paradigm of iterative refinement, inspiring many extensions targeting efficiency, aggregation, and memory [18, 57, 58, 62, 69]. To better capture long-range dependencies, Transformer-based approaches integrate global matching and the attention mechanism [17, 18, 47, 63, 64], while alternative formulations explore using diffusion models [30, 43]. However, two-frame methods inherently lack temporal context. To address this, multi-frame architectures incorporate temporal windows [4, 46], memory buffers [10, 26, 51] or accumulate correspondences over longer horizons [14, 61]. Despite these advances, both two-frame and multi-frame frameworks typically train task-specific feature extractors and depend heavily on per-domain fine-tuning, which severely limits their zero-shot generalization across diverse, real-world environments.

Vision Transformers for Optical Flow. Large pre-trained vision Transformers capture exceptionally rich representations that transfer across diverse domains [22, 37,48,54,60,65]. For optical flow, existing approaches typically exploit these models by either directly regressing flow via lightweight heads [43, 59, 66] or by designing task-specific blocks to process local cost volumes and iterative updates [17, 30, 47, 57, 70]. While these strategies improve feature robustness, confining powerful static priors to the local search space or unconstrained regression inherently bottlenecks their ability to resolve extreme displacements.

Tracking Any Point. The Tracking Any Point (TAP) task [8, 42] evaluates a model's ability to maintain robust, long-range temporal correspondences. Traditionally, optical flow has served as a building block for multi-frame TAP problems [9, 36, 68]. To better handle occlusions and out-of-frame motions, recent methods [6, 20, 21] exploit correlations across multiple simultaneous tracks. Concurrently, another emerging line of research demonstrates that leveraging the rich, geometric-aware features from visual foundation models significantly enhances long-term tracking robustness [1–3]. Pushing towards unified dense tracking, recent models like AllTracker [14] and a concurrent method CoWtracker [24] propose a single-model solution using iterative warping-based refinement. Yet, their reliance on local search paradigms inevitably leads to error accumulation and compromised pixel-level accuracy over time. In contrast, by leveraging generalized global matching and local refinement, MegaFlow naturally excels at both point tracking and optical flow within a single unified architecture.

3 Method

Given a sequence of T video frames $\{I_1, I_2, \ldots, I_T\}$, our goal is to estimate the optical flows $\{f_1, \ldots, f_{T-1}\} \in \mathbb{R}^{H\times W\times 2}$ between consecutive frames. Specifically, our framework consists of three core components. First, all frames are processed

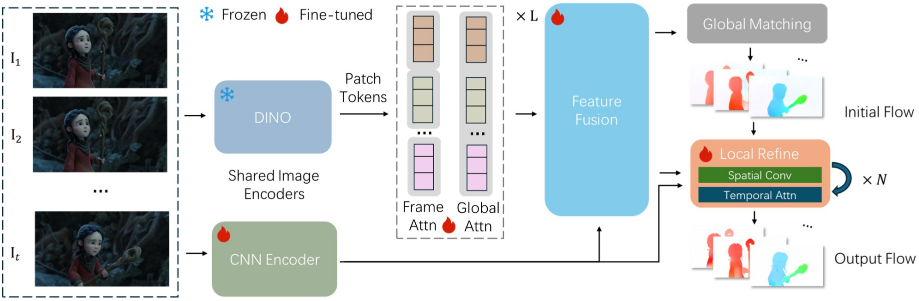


Fig. 2: MegaFlow. Given an input sequence, a frozen DINO and a trainable CNN extract dense patch tokens and local structural features. Alternating frame and global attention, followed by feature fusion, process these tokens into a globally consistent representation. Pair-wise global matching then computes initial flows. Finally, a recurrent module iteratively refines the initial flows using spatial convolutions and temporal attention for sub-pixel accuracy. Crucially, our design seamlessly processes variable-length inputs without architectural modifications.

by a shared feature extractor that integrates a trainable local CNN with a frozen vision Transformer backbone. This produces multi-frame feature maps $\{F_1, \ldots, F_T\}$ that capture both fine-grained local structures and robust, globally consistent representations. Second, we perform pair-wise global matching between adjacent feature maps to compute an initial flow estimation. Finally, a recurrent refinement module iteratively updates these flow fields using local correlation and temporal attention, yielding accurate and temporally consistent optical flows. An overview of the full architecture is shown in Fig. 2.

3.1 Feature Extraction and Fusion

Given an input sequence of T frames $\{I_1, \ldots, I_T\}$, we first extract per-frame features using DINOv2 [37,48]. This produces a set of patch tokens $\{t_i\}_{i=1}^{T}$, which are subsequently processed by a Transformer backbone composed of L alternating frame-wise and global self-attention layers following VGGT [54].

To compensate for the loss of fine spatial details caused by patch tokenization, we introduce a lightweight CNN encoder [15] to produce multi-scale feature maps at 1/2 and 1/4 resolutions. The 1/4 resolution features are spatially compressed using pixel unshuffle [45] and concatenated with the intermediate Transformer tokens. A DPT-style fusion head [39] then aligns and merges the CNN features with the Transformer feature space, producing the fused multi-frame feature maps $\{F_i \in \mathbb{R}^{\frac{H}{7} \times \frac{W}{7} \times D}\}_{i=1}^{T}$, where $D = 128$. These fused features retain strong local details while encoding broad cross-frame context, serving as the input for the global matching and refinement stages.

3.2 Global Matching

With the fused feature maps $\{F_i\}_{i=1}^{T}$, we estimate an initial flow between each adjacent pair (F_i, F_{i+1}). We formulate correspondence as a global matching problem. For each spatial location $\mathbf{u}$ in F_i, we compare it against all locations in F_{i+1} by computing an all-pairs correlation:

$$C_i(\mathbf{u}, \mathbf{v}) = \langle F_i(\mathbf{u}),\, F_{i+1}(\mathbf{v}) \rangle, \quad C_i \in \mathbb{R}^{\frac{H}{7} \times \frac{W}{7} \times \frac{H}{7} \times \frac{W}{7}}, \tag{1}$$

where $\langle \cdot, \cdot \rangle$ represents dot product, $\mathbf{u}, \mathbf{v}$ denote pixels. Following [63, 64], we apply a softmax normalization over $\mathbf{v}$ to obtain a probability distribution over correspondence candidates:

$$M_i(\mathbf{u}, \mathbf{v}) = \mathrm{softmax}_{\mathbf{v}}(C_i(\mathbf{u}, \mathbf{v}))\,, \tag{2}$$

Let G denote the coordinate grid of F_{i+1}. The matched coordinate field is then computed as an expectation over this distribution, and the initial flow is simply the displacement between source and matched coordinates:

$$f_i^{\mathrm{init}}(\mathbf{u}) = \sum_{\mathbf{v}} M_i(\mathbf{u}, \mathbf{v}) \cdot G(\mathbf{v}) \;-\; G(\mathbf{u}). \tag{3}$$

3.3 Local Recurrent Refinement

After obtaining the initial flows $\{f_i^{\mathrm{init}}\}_{i=1}^{T-1}$, we perform iterative refinement using a lightweight recurrent module that jointly leverages local spatial correlations and temporal dependencies.

For each adjacent frame pair $(i, i{+}1)$, the initial flow is bilinearly upsampled to the CNN feature resolution, $\tilde{f}_i^{\mathrm{init}}$, and used to sample the CNN feature of I_{i+1}. A local correlation volume is then constructed as:

$$C_i^{\mathrm{local}}(\mathbf{u}) = \left\langle F_i^{\mathrm{cnn}}(\mathbf{u}), F_{i+1}^{\mathrm{cnn}}\left(\mathbf{u} + \tilde{f}_i^{\mathrm{init}}(\mathbf{u}) + \Delta\mathbf{u}\right) \right\rangle, \tag{4}$$

where $\mathbf{u}$ denotes a pixel location at 1/4 resolution and $\Delta\mathbf{u} \in [-r, r]^2$ enumerates local offsets.

The refinement is performed iteratively for K steps, where each update follows:

$$f_i^{(k+1)} = f_i^{(k)} + \mathcal{R}\left(f_i^{(k)}, C_i^{\mathrm{local}}, F_i^{\mathrm{cnn}}, F_i\right), \tag{5}$$

with $\mathcal{R}$ denotes the refinement network and $f_i^{(0)} = \tilde{f}_i^{\mathrm{init}}$.

Specifically, $\mathcal{R}$ consists of two complementary components: (1) a ConvNeXt-based convolutional branch aggregates local motion evidence from the correlation and CNN features, and (2) a temporal attention branch correlates features across the sequence. This hybrid formulation enables the model to capture both fine-grained local motion and long-range temporal coherence, leading to robust flow estimation under occlusions and appearance changes.

Finally, the refinement operates recurrently over all frames, enforcing spatial consistency and temporal smoothness across the sequence. The design remains agnostic to the number of input frames, allowing MegaFlow to generalize to variable-length sequences without structural modification.

3.4 Training Loss

The model is trained with the following objective for all $T-1$ output flows:

$$\mathcal{L}_{\text{flow}} = \sum_{i}^{T-1} \left\| f_i^{\text{init}} - \hat{f}_i \right\|_{\text{smooth}} + \sum_{k=1}^{K} \gamma^{K-k} \sum_{i}^{T-1} \left\| f_i^k - \hat{f}_i \right\|_1, \tag{6}$$

where $\hat{f}$ represents ground-truth optical flow, $\|\cdot\|_{\text{smooth}}$ denotes the smooth ℓ_1 loss, K is the iteration number, $\gamma = 0.9$ is the weight. We apply exponentially increasing weights [52] to supervise the iteratively refined flow.

3.5 Extension to Point Tracking

A key advantage of MegaFlow is that it seamlessly adapts to dense point tracking without requiring any architectural modifications. Given a video sequence of $T+1$ frames, denoted as $\{I_0, I_1, \ldots, I_T\}$, we define I_0 as the query frame and the subsequent frames $\{I_1, \ldots, I_T\}$ as the targets. Let $\mathbf{x} \in \mathbb{R}^2$ denote a 2D pixel location in image coordinate, and let $\mathcal{P} \subset \mathbb{R}^2$ represent the subset of query locations we aim to track. We operate under the standard assumption that the initial position is anchored at the query coordinate, such that $p_0(\mathbf{x}) = \mathbf{x}$.

Our objective is to estimate the corresponding location $p_t(\mathbf{x}) \in \mathbb{R}^2$ for every query pixel $\mathbf{x} \in \mathcal{P}$ across all target frames I_t, where $t \in \{1, \ldots, T\}$. To bridge the task of dense tracking with our multi-frame flow formulation, we parameterize the trajectories using the displacement field. Instead of calculating flow strictly for adjacent frames, our global matching and local refinement stages explicitly compute correspondences $f_{0\to t}$ between the first frame I_0 and each target frame I_t. The tracked position of any pixel is directly derived as:

$$p_t(\mathbf{x}) = \mathbf{x} + f_{0\to t}(\mathbf{x})$$

In our dense tracking scenario, we define $\mathcal{P}$ to encompass the entire spatial grid of the query frame, effectively tracking all points simultaneously. To optimize the network using sparse point track supervision, we adapt our training objective to directly penalize trajectory errors rather than dense flow fields. The tracking loss is formulated as:

$$\mathcal{L}_{\text{point}} = \sum_{t=1}^{T} \left\| p_t^{\text{init}} - \hat{p}_t \right\|_{\text{smooth}} + \sum_{k=1}^{K} \gamma^{K-k} \sum_{t=1}^{T} \left\| p_t^k - \hat{p}_t \right\|_1, \tag{7}$$

where $\hat{p}_t$ represents the ground truth tracked point coordinates at time step t.

To process long sequences, we employ an sliding window strategy with a window size of 8. Specifically, we use the predicted trajectories from the current window as the initialization for the next, sequentially repeating the inference process following [14]. Unlike dedicated point trackers that depend on explicit visibility heuristics and confidence scores, MegaFlow propagates tracks entirely through its continuous displacement fields.

Table 1: Zero-shot evaluation on Sintel (train) and KITTI (train). Most methods are trained on the FlyingChairs and FlyingThings datasets by default. Note that VideoFlow models [46] are not trained on FlyingChairs and UFM [66] are trained on multiple dense correspondence datasets.

Method	Frames	Sintel (train)		KITTI (train)	
		Clean↓	Final↓	Fl-epe↓	Fl-all↓
PWC-Net [50]	2	2.55	3.93	10.4	33.7
RAFT [52]		1.43	2.71	5.04	17.4
GMA [18]		1.30	2.74	4.69	17.1
DIP [69]		1.30	2.82	4.29	13.7
RPKNet [35]		1.12	2.45	-	13.0
GMFlow [63]		1.08	2.48	11.20	28.7
FlowFormer [17]		1.01	2.40	4.09	14.7
Flowformer++ [47]		0.90	2.30	3.93	14.2
SEA-RAFT (L) [58]		1.19	4.11	3.62	12.9
AnyFlow [19]		1.10	2.52	3.76	12.4
SAMFlow [70]		0.87	2.11	3.44	12.3
FlowDiffuser [30]		0.86	2.19	3.61	11.8
WAFT-DINOv3-a2 [57]		1.28	2.56	3.49	12.9
UFM [66]		1.15	2.01	**2.96**	11.0
MegaFlow		0.89	2.07	3.00	**10.6**
VideoFlow-BOF* [46]	3	1.03	2.19	3.96	15.3
VideoFlow-MOF* [46]	5	1.18	2.56	3.89	14.2
StreamFlow [51]	4	0.87	2.11	3.85	12.6
MemFlow [10]	3	0.93	2.08	3.88	13.7
MemFlow-T [10]	3	**0.85**	2.06	3.38	12.8
MegaFlow	4	**0.85**	**1.83**	3.20	10.7

4 Experiments

Datasets and Evaluation Protocol. We first train on FlyingChairs [11], TartanAirV1 [55], and FlyingThings [32], following standard practices [4, 52, 58, 63, 64]. Zero-shot evaluation is then conducted on Sintel [5] and KITTI [12] to assess cross-domain generalization. In addition, we train on a mixed dataset comprising FlyingThings [32], HD1K [23], Sintel [5], and KITTI [12], and report results on online benchmarks of Sintel, KITTI, and Spring [33]. Importantly, we do not perform dataset-specific fine-tuning for benchmark submissions.

Evaluation Metrics. We follow standard optical flow evaluation metrics. The primary metric is End-Point Error (EPE), calculated as the average ℓ_2 distance between the predicted and ground-truth flow. For KITTI, we additionally report F1-all, indicating the percentage of outliers. For the Spring dataset, we include 1-pixel outlier rate (1px), percentage of flow outliers (Fl), and weighted area under the curve (WAUC) [12, 33, 41]. To further analyze performance across motion scales, we report EPE by flow magnitude: s_{0-10}, s_{10-40}, and s_{40+} correspond to flow magnitudes of 0–10, 10–40, and over 40 pixels, respectively.

Implementation Details. The Transformer backbone consists of $L = 24$ layers of alternating global and frame-wise attention following VGGT [54]. Local

Table 2: Optical flow estimation across different motion magnitudes. MegaFlow significantly reduces EPE on extreme large displacements (s_{40+}).

Method	Sintel (Clean) ↓			Sintel (Final) ↓		
	s_{40+}	s_{10-40}	s_{0-10}	s_{40+}	s_{10-40}	s_{0-10}
SEA-RAFT [58]	8.286	1.343	0.261	26.878	4.259	0.547
MemFlow-T [10]	5.239	0.980	**0.211**	13.670	2.224	0.371
SAMFlow [70]	5.117	1.029	0.215	13.575	2.429	0.396
UFM [66]	6.836	1.209	0.259	12.963	2.129	0.399
WAFT-DINOv3-a2 [57]	8.870	1.218	0.217	13.192	1.966	**0.324**
MegaFlow	**4.729**	**0.909**	0.314	**11.175**	**1.941**	0.480

features are extracted using the first two blocks of a ResNet [15] pretrained on ImageNet [7], producing 1/4 resolution feature maps. The refinement module comprises two ConvNeXt blocks [27, 58] and two temporal attention blocks. The full model contains 936M parameters.

We implement our model with PyTorch [38] and optimize with AdamW [28]. The DINOv2 [37] image encoder, Transformer blocks and parts of the feature fusion module are initialized with pre-trained VGGT [54] weights, and the DINOv2 is kept frozen during training. Training proceeds in 3 stages: (1) 20K iterations on FlyingChairs; (2) 30K iterations on TartanAirV1; (3) 30K iterations on FlyingThings, split into 15K iterations of 2-frame pretraining followed by 15K iterations of multi-frame training; (4) 20K iterations on the mixed dataset. The batch size is set to 128 for all stages. The refinement module uses 4 iterations during training and 8 during evaluation. During multi-frame training, we randomly sample between 2 and 6 frames from a random scene, encouraging the model to handle variable temporal spans. By default, inference uses $T = 4$ input frames.

The full training process runs on 64 NVIDIA GH200 GPUs over four days. We employ gradient norm clipping with a threshold of 1.0 to ensure stability, and leverage bfloat16 precision and gradient checkpointing to improve memory usage and computational efficiency. All attention layers are accelerated using FlashAttention-3 [44]. Additional details are provided in the supplementary.

4.1 Zero-Shot Generalization

We evaluate the zero-shot performance of MegaFlow on the Sintel and KITTI training sets after the third stage training. As shown in Tab. 1, MegaFlow establishes a new state-of-the-art, outperforming both two-frame and multi-frame architectures. On Sintel, while maintaining a highly competitive Clean EPE of 0.85, MegaFlow significantly advances the Final EPE to an unprecedented 1.83. Because the Sintel Final pass introduces severe motion blur, atmospheric effects, and complex occlusions, this substantial margin demonstrates the exceptional robustness of MegaFlow against extreme appearance changes.

Furthermore, on KITTI benchmark, MegaFlow achieves the best overall Fl-all error of 10.7 and a highly competitive Fl-epe of 3.00. Notably, MegaFlow overall

Table 3: Zero-shot and fine-tuned point tracking comparison on TAP-Vid [8] benchmarks. We evaluate δ_{avg} using an input resolution of 384 × 512. Note that the optical flow models are trained solely on mixed optical flow datasets, while point trackers are trained on tracking datasets. Despite this, our zero-shot flow model achieves competitive performance comparable to dedicated trackers. After fine-tuning ('Flow → Kubric'), MegaFlow establishes state-of-the-art results.

Method	Training	DAVIS ↑	Kinetics ↑	RGB-Stacking ↑	Mean ↑
PIPs++ [68]	PointOdyssey	62.5	64.2	70.4	65.7
LocoTrack [6]	Kubric	68.0	70.0	80.3	72.8
BootsTAPIR [9]	Kubric+15M	67.9	70.6	81.0	73.1
CoTracker2 [21]	Kubric	70.9	65.8	73.4	70.0
CoTracker3-Kub [20]	Kubric	77.4	70.6	83.4	77.1
CoTracker3 [20]	Kubric+15K	77.1	71.8	84.2	77.7
AllTracker-Kub [14]	Kubric	75.2	71.3	90.1	78.9
AllTracker [14]	Kubric+mix	76.3	**72.3**	90.0	79.5
MegaFlow	Flow → Kubric	**77.6**	70.2	**91.1**	**79.6**
RAFT [52]	Flow mix	48.5	64.3	82.8	65.2
SEA-RAFT [58]	Flow mix	48.7	64.3	85.7	66.2
AccFlow [61]	Flow mix	23.5	38.8	63.2	41.8
MemFlow-T [10]	Flow mix	61.7	64.9	85.2	70.6
WAFT-DINOv3-a2 [57]	Flow mix	53.9	61.0	84.0	66.3
MegaFlow	Flow mix	**65.6**	**65.5**	**89.6**	**73.6**

Table 4: Benchmark comparison of optical flow methods on the Spring test set.

	Method	Frames	EPE ↓	Fl ↓	WAUC ↑	1px ↓
w/o Fine-tune	PWC-Net [50]	2	2.288	4.889	45.670	82.265
	RAFT [52]	2	1.476	3.198	90.920	6.790
	GMA [18]	2	0.914	3.079	90.722	7.074
	FlowFormer [17]	2	0.723	2.384	91.679	6.510
	RPKNet [35]	2	0.657	1.756	92.638	4.809
	MemFlow [10]	3	0.627	2.114	92.253	5.759
	StreamFlow [51]	4	0.606	1.856	93.253	5.215
	MegaFlow	2	0.403	1.451	93.502	**4.432**
	MegaFlow	4	**0.349**	**1.261**	**93.738**	4.521
w/ Fine-tune	CrocoFlow [59]	2	0.498	1.508	93.660	4.565
	SEA-RAFT (M) [58]	2	0.363	1.347	94.534	3.686
	MemFlow [10]	3	0.471	1.416	93.855	4.482
	StreamFlow [51]	4	0.467	1.424	94.404	4.152
	WAFT-DINOv3-a2 [57]	2	**0.325**	**1.246**	**95.051**	**3.289**

outperforms recent models with similar vision priors like UFM [66], despite UFM was trained on significantly more diverse dense correspondence datasets. Overall, the zero-shot evaluation confirms that our approach balances precision and generalization, effectively handling diverse motion dynamics across both synthetic and real-world benchmarks.

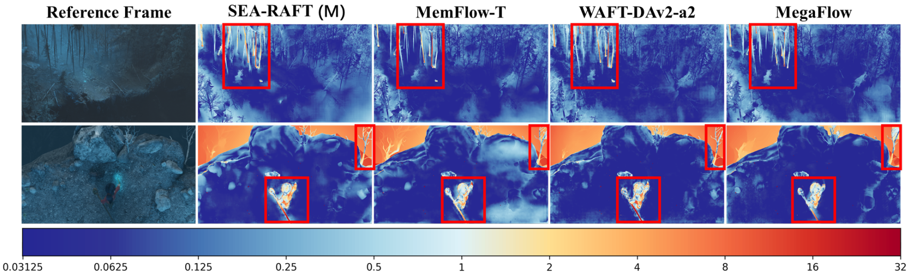


Fig. 3: Qualitative comparison of optical flow. Visualization of SEA-RAFT [58], MemFlow [10], WAFT-DAv2-a2 [57], and our method on the Spring benchmark. The colorbar indicates endpoint error. Our approach outperforms prior methods, demonstrating that our method generalizes well to Full HD resolution while preserving both local and global motion details.

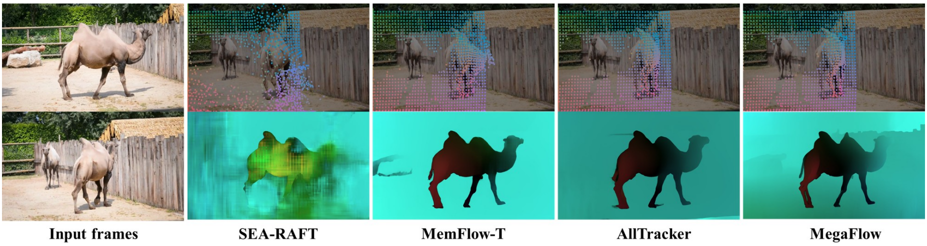


Fig. 4: Qualitative comparison of long-range point tracking. Visualization of SEA-RAFT [58], MemFlow [10] and AllTracker [14] and our method on the DAVIS benchmark. The first column shows the input frames (spanning 90 frames). The top row visualizes long-range dense point tracking, while the bottom row shows the corresponding optical flow between the first and last frame. Our method produces more accurate and temporally consistent tracks and flow estimates over very long sequences.

4.2 Large Displacement Optical Flow

To explicitly assess robustness against extreme motion, we evaluate MegaFlow in a zero-shot setting across different flow magnitude intervals on the Sintel dataset (Tab. 2). Following the evaluation protocol from Sec. 4.1, we compare our approach against state-of-the-art two-frame methods [57, 58, 66, 70] and the multi-frame architecture MemFlow-T [10].

As illustrated by the error curves in Fig. 1(a), the End-Point Error (EPE) of baseline methods escalates rapidly as motion magnitude increases. The quantitative breakdown in Tab. 2 further confirms this vulnerability: although recent iterative-based baselines perform competitively on minor displacements (s_{0-10}), their accuracy degrades severely in the extreme s_{40+} regime. While multi-frame models like MemFlow-T attempt to mitigate this by accumulating sequential memory, they remain inherently bottlenecked by localized search paradigms. In contrast, MegaFlow effectively flattens this error curve and establishes a significant margin of improvement on extreme displacements, driving the s_{40+} error down to a remarkable 4.729 on Sintel (Clean) and 11.175 on Sintel (Final).

4.3 Long-Range Point Tracking

To assess representation robustness on long-range correspondences, we evaluate MegaFlow on TAP-Vid [8] benchmarks (including Kinetics, DAVIS, RGB-Stacking datasets) *without architectural modifications.* For zero-shot evaluation, we applying the procedure in Sec. 3.5 to all flow-based baselines. To establish our architecture's full tracking potential, we evaluate a variant fine-tuned on Kubric [13] for 20K iterations after trained on mixed flow dataset. Notably, our sliding window approach processes up to 600-frame sequences on a single GH200 GPU.

Following standard protocols, we evaluate all models at 384×512 resolution using the δ_{avg} metric [14,20,21], which averages $\delta_k = 100 \cdot \mathbf{1}[\|p - \hat{p}\|_2 < k]$ across $k \in \{1, 2, 4, 8, 16\}$. Baselines include state-of-the-art trackers [6,9,20,21,68] and flow-based methods [10,52,57,58,61].

As shown in Tab. 3 and Fig. 1(b), MegaFlow exhibits exceptional cross-task transferability. In the strictly zero-shot setting, it achieves a 73.6% average accuracy, significantly outperforming all flow baselines and surpassing dedicated trackers like PIPs++, LocoTrack, and CoTracker2. On RGB-Stacking, our zero-shot model exceeds nearly all task-specific architectures. Fine-tuned solely on Kubric ('Flow → Kubric'), MegaFlow establishes a new state-of-the-art average of 79.6%, decisively outperforming CoTracker3 and AllTracker despite their exposure to more extensive tracking datasets.

Qualitatively (Fig. 4), MegaFlow consistently yields accurate, coherent dense tracks. Unlike local search methods (e.g., SEA-RAFT) that produce unstable trajectories on long-range motions, or MemFlow which suffers from boundary artifacts around articulated limbs, our global matching architecture remains robust. Furthermore, while AllTracker achieves reasonable coherence on dense grids, it lacks pixel-level accuracy. Conversely, MegaFlow maintains sharp local structures across extended sequences, successfully bridging long-range stability with high-fidelity flow estimation.

Ultimately, our unified flow formulation generalizes exceptionally well to long-range tracking. The zero-shot performance proves that foundation vision priors and global matching transfer effectively without tracking-specific supervision, while fine-tuning confirms MegaFlow as a highly capable architecture for generalized dense motion estimation. Additional in-the-wild visual results are provided in the supplementary material.

4.4 Benchmark Results

Spring. Without fine-tuning, MegaFlow achieves state-of-the-art zero-shot performance on the Spring benchmark, obtaining the lowest EPE and Fl-all scores in Tab. 4. Notably, it surpasses most models specifically fine-tuned on high-resolution (1080×1920) data, demonstrating robust generalization despite being trained solely on standard-resolution datasets (432×960). This scalability stems from our global initialization, which provides a strong cross-frame prior, and a hybrid refinement module that effectively integrates spatial and cross-frame cues across varying resolutions. While WAFT [57] exhibits marginally higher accuracy through high-resolution supervision, MegaFlow remains highly competitive

Table 5: Benchmark results on Sintel (test) and KITTI (test). Methods are separated into two-frame and multi-frame models. We explicitly report results on large-displacement motions (s_{40+}) for the Sintel benchmark.

Method	Sintel (Clean) ↓		Sintel (Final)↓		KITTI ↓
	EPE	s_{40+}	EPE	s_{40+}	Fl-all
PWC-Net [50]	3.86	25.29	5.04	31.07	9.60
RAFT [52]	1.61	9.28	2.86	16.37	5.10
GMA [18]	1.39	7.60	2.47	13.50	5.15
SEA-RAFT (L) [58]	1.31	7.90	2.60	15.71	4.30
FlowFormer [17]	1.16	6.44	2.09	11.67	4.68
RPKNet [35]	1.32	7.28	2.66	16.00	4.64
CrocoFlow [59]	1.09	6.30	2.44	15.21	3.64
DDVM [43]	1.75	12.22	2.48	16.55	**3.26**
AnyFlow [19]	1.21	7.32	2.44	14.20	4.41
SAMFlow [70]	1.00	5.25	2.08	11.28	4.49
FlowDiffuser [30]	1.02	5.57	2.03	10.93	4.17
WAFT-DINOv3-a2 [57]	0.95	5.52	2.02	12.49	3.56
VideoFlow-BOF [46]	1.01	5.61	1.71	9.42	4.44
VideoFlow-MOF [46]	0.99	5.48	**1.65**	**8.80**	3.65
StreamFlow [51]	1.04	6.00	1.87	10.68	4.24
MemFlow-T [10]	1.08	6.02	1.83	9.83	3.88
MegaFlow	**0.91**	**4.84**	2.43	15.50	3.71

without any dataset-specific adaptation, underscoring its inherent architectural scalability.

Qualitative results (Fig. 3) show that MegaFlow produces sharper motion boundaries and preserves finer details than prior methods, particularly for thin structures such as tree branches and slender objects.

Sintel and KITTI. As shown in Tab. 5, MegaFlow achieves competitive performance using a *single, unified model* without dataset-specific fine-tuning. On Sintel (Clean), we reach a state-of-the-art EPE of 0.91 and exhibit superior robustness on extreme displacements (s_{40+}) with an EPE of 4.84. While many specialized methods rely on per-benchmark adaptation, our approach prioritizes the general-purpose motion representations inherent in foundation models. Consistent with [43,57], we observe that the 'Ambush 1' sequence in Sintel (Final) remains a significant outlier. As detailed in the supplementary material, excluding this anomalous sequence reveals that MegaFlow outperforms recent architectures such as WAFT and MemFlow, while achieving results fully comparable to the VideoFlow framework. This confirms our model's superior capability in handling complex, occluded motion. On KITTI, MegaFlow yields a competitive Fl-all. This performance is primarily influenced by the fixed-patch tokenization of the Transformer backbone, which is sensitive to KITTI's specific resolution and aspect ratio. Unlike baselines employing exhaustive multi-scale inference or specialized padding, we maintain a zero-shot-style evaluation to verify intrinsic transferability. In this context, MegaFlow remains on par with other architectures with pretrained prior [57,59,70], effectively balancing high precision with broad generalization. See supplementary for more qualitative benchmark results.

Table 6: Ablation of Pre-trained Priors and Architecture. Evaluated zero-shot using 2 input frames. The inference latency is measured on an RTX 4090 at 540 × 960.

	Variant	Param(M)	Latency (ms)	Memory (GB)	S-Clean ↓	S-Final ↓	K-epe ↓	K-all ↓
	6 layers	256	140.4	2.43	2.05	2.46	9.71	17.9
w/o Pre-train	12 layers	483	189.5	3.56	2.13	3.31	11.70	19.8
	Conv embedder	632	262.1	4.95	1.05	2.38	4.89	15.4
	6 layers	256	140.4	2.43	1.22	2.30	7.14	16.1
	12 layers	483	189.5	3.56	1.11	2.23	6.76	14.4
w/ Pre-train	Freeze transformer	936	323.9	6.08	1.22	3.17	5.13	14.7
	w/o feat fusion	935	321.1	5.81	1.02	2.11	4.71	13.9
	Full Model	936	323.9	6.08	**0.89**	**2.08**	**3.00**	**10.9**

4.5 Ablation Study

To analyze the contributions of the pretrained vision prior, architectural adaptations, and temporal reasoning, we conduct comprehensive ablations evaluated zero-shot on Sintel and KITTI. All variants are trained under identical zero-shot settings above to ensure fair comparisons. We also report parameter counts, latency, and peak memory to provide full transparency on computational trade-offs.

The Role of Priors and Scale. Tab. 6 shows that scaling transformer depth L from 6 to 12 layers from scratch actually degrades zero-shot performance. This indicates that without proper initialization, increasing model capacity exacerbates optimization difficulties on large displacements. However, introducing the VGGT pretrained prior effectively regularizes the training and unlocks the ability to scale up the architecture, enabling our 24-layer full model to fully leverage its massive capacity and achieve optimal performance. Meanwhile, the image encoding strategy impacts motion estimation. While a standard convolutional embedder provides reasonable local features, it lacks the aligned semantic space inherent in foundation models.

Feature Fusion and Fine Tuning. Freezing the backbone or removing the CNN feature fusion causes severe performance drops (e.g., Fl-all 10.9 to 14.7). While the prior provides a global geometric information, fine tuning and structural fusion remain essential to recover the local details.

Multi-Frame Consistency and Temporal Attention. As shown in Fig. 5, evaluating frames in isolated pairs (middle row) yields temporally inconsistent predictions and localized artifacts around occluders. By leveraging multi-frame context (bottom row), MegaFlow ensures highly stable flow estimations. Tab. 7 quantifies this advantage. On the complex Sintel benchmark, expanding the context window to 4 frames drastically reduces the Final EPE from 2.08 to 1.83. Crucially, this improvement strictly relies on our temporal attention module. Without it, the network struggles to effectively aggregate the extended context, yielding marginal gains.

Conversely, KITTI performance degrades as frame count increases. KITTI features rapid forward ego-motion where objects quickly exit the field of view. In these extreme scenarios, extended temporal windows introduce boundary occlusion artifacts rather than useful structural priors. Consequently, our architecture elegantly supports flexible inputs, defaulting to a 4-frame context for complex general motion while maintaining an optimal 2-frame setup for rapid ego-motion.

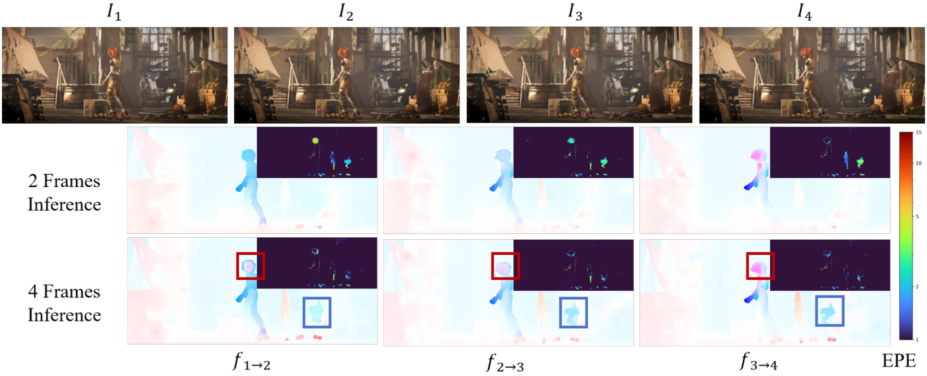


Fig. 5: Impact of multi-frame context on temporal consistency. Top row: Consecutive input frames. Middle row: Optical flow estimated from isolated frame pairs. Bottom row: Flow estimated jointly ($T = 4$). Processing isolated pairs leads to temporal inconsistencies and occlusion artifacts, particularly around the moving subject (red boxes) and background structures (blue boxes). In contrast, our expanded multi-frame context produces highly stable and accurate motion boundaries.

Table 7: Ablation of Temporal Attention. Evaluated zero-shot across varying input frame counts ($T \in \{2, 4, 6\}$). Expanding the context window to $T = 4$ optimally resolves complex motion blur on Sintel, while a standard 2-frame setup avoids occlusion noise on the fast, forward-driving KITTI dataset.

Method	Sintel Clean ↓			Sintel Final ↓			KITTI Fl-epe ↓		
	2	4	6	2	4	6	2	4	6
w/o temporal attn	**0.88**	0.95	0.98	2.09	1.99	2.04	3.12	4.22	4.56
w/ temporal attn	0.89	**0.85**	**0.94**	**2.08**	**1.83**	**1.92**	**3.00**	**3.44**	**3.86**

5 Conclusion

We presented MegaFlow, a unified architecture designed for large displacement motion estimation. Our approach demonstrates that integrating pretrained foundation vision priors with a synergistic global and local feature formulation effectively addresses the challenge of large displacements. Extensive evaluations show that MegaFlow achieves state-of-the-art zero-shot performance in optical flow. Crucially, this robust representation exhibits remarkable cross-task transferability. Without any architectural modifications, MegaFlow delivers zero-shot point tracking that rivals dedicated pipelines, and establishes new state-of-the-art results after fine-tuning on point tracking.

Limitations and Future Work. While MegaFlow demonstrates strong generalization, dense multi-frame modeling inherently increases the computational overhead for longer sequences. Future work will focus on improving sequence-level efficiency and exploring unified pre-training paradigms to jointly optimize dense optical flow and long-range tracking. Ultimately, MegaFlow represents a promising step toward building a robust and generalized foundation for extreme motion estimation across diverse real-world applications.

References

1. Aydemir, G., Cai, X., Xie, W., Güney, F.: Track-on: Transformer-based online point tracking with memory. In: The Thirteenth International Conference on Learning Representations (2025) 3
2. Aydemir, G., Xie, W., Güney, F.: Can visual foundation models achieve long-term point tracking? arXiv preprint arXiv:2408.13575 (2024) 3
3. Aydemir, G., Xie, W., Güney, F.: Track-on2: Enhancing online point tracking with memory. arXiv preprint arXiv:2509.19115 (2025) 3
4. Bargatin, V., Chistov, E., Yakovenko, A., Vatolin, D.: Memfof: High-resolution training for memory-efficient multi-frame optical flow estimation. arXiv preprint arXiv:2506.23151 (2025) 3, 7
5. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: ECCV. pp. 611–625. Springer (2012) 7
6. Cho, S., Huang, J., Nam, J., An, H., Kim, S., Lee, J.Y.: Local all-pair correspondence for point tracking. arXiv preprint arXiv:2407.15420 (2024) 3, 9, 11
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009) 8
8. Doersch, C., Gupta, A., Markeeva, L., Recasens, A., Smaira, L., Aytar, Y., Carreira, J., Zisserman, A., Yang, Y.: Tap-vid: A benchmark for tracking any point in a video. Advances in Neural Information Processing Systems **35**, 13610–13626 (2022) 3, 9, 11
9. Doersch, C., Luc, P., Yang, Y., Gokay, D., Koppula, S., Gupta, A., Heyward, J., Rocco, I., Goroshin, R., Carreira, J., et al.: Bootstap: Bootstrapped training for tracking-any-point. In: Proceedings of the Asian Conference on Computer Vision. pp. 3257–3274 (2024) 3, 9, 11
10. Dong, Q., Fu, Y.: Memflow: Optical flow estimation and prediction with memory. In: CVPR. pp. 19068–19078 (2024) 3, 7, 8, 9, 10, 11, 12
11. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: ICCV. pp. 2758–2766 (2015) 2, 3, 7
12. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: CVPR. pp. 3354–3361. IEEE (2012) 1, 7
13. Greff, K., Belletti, F., Beyer, L., Doersch, C., Du, Y., Duckworth, D., Fleet, D.J., Gnanapragasam, D., Golemo, F., Herrmann, C., et al.: Kubric: A scalable dataset generator. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3749–3761 (2022) 11
14. Harley, A.W., You, Y., Sun, X., Zheng, Y., Raghuraman, N., Gu, Y., Liang, S., Chu, W.H., Dave, A., Tokmakov, P., You, S., Ambrus, R., Fragkiadaki, K., Guibas, L.J.: AllTracker: Efficient dense point tracking at high resolution. In: ICCV (2025) 3, 6, 9, 10, 11
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016) 4, 8
16. Horn, B.K., Schunck, B.G.: Determining optical flow. Artificial intelligence **17**(1-3), 185–203 (1981) 2, 3
17. Huang, Z., Shi, X., Zhang, C., Wang, Q., Cheung, K.C., Qin, H., Dai, J., Li, H.: Flowformer: A transformer architecture for optical flow. In: ECCV. pp. 668–685. Springer (2022) 2, 3, 7, 9, 12

18. Jiang, S., Campbell, D., Lu, Y., Li, H., Hartley, R.: Learning to estimate hidden motions with global motion aggregation. In: ICCV. pp. 9772–9781 (2021) 2, 3, 7, 9, 12
19. Jung, H., Hui, Z., Luo, L., Yang, H., Liu, F., Yoo, S., Ranjan, R., Demandolx, D.: Anyflow: Arbitrary scale optical flow with implicit neural representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5455–5465 (2023) 7, 12
20. Karaev, N., Makarov, I., Wang, J., Neverova, N., Vedaldi, A., Rupprecht, C.: CoTracker3: Simpler and better point tracking by pseudo-labelling real videos (2024) 3, 9, 11
21. Karaev, N., Rocco, I., Graham, B., Neverova, N., Vedaldi, A., Rupprecht, C.: Cotracker: It is better to track together. In: European conference on computer vision. pp. 18–35. Springer (2024) 3, 9, 11
22. Keetha, N., Müller, N., Schönberger, J., Porzi, L., Zhang, Y., Fischer, T., Knapitsch, A., Zauss, D., Weber, E., Antunes, N., et al.: Mapanything: Universal feed-forward metric 3d reconstruction. arXiv preprint arXiv:2509.13414 (2025) 2, 3
23. Kondermann, D., Nair, R., Honauer, K., Krispin, K., Andrulis, J., Brock, A., Gussefeld, B., Rahimimoghaddam, M., Hofmann, S., Brenner, C., 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. pp. 19–28 (2016) 2, 7
24. Lai, Z., Insafutdinov, E., Sucar, E., Vedaldi, A.: Cowtracker: Tracking by warping instead of correlation. arXiv preprint arXiv:2602.04877 (2026) 3
25. Lin, C., Lin, Y., Pan, P., Yu, Y., Yan, H., Fragkiadaki, K., Mu, Y.: Movies: Motion-aware 4d dynamic view synthesis in one second. arXiv preprint arXiv:2507.10065 (2025) 2
26. Liu, J., Liu, M., Zhu, S., Zhang, Y., Li, J., Yang, M.Y., Nex, F., Cheng, H., Wang, H.: Arflow: Auto-regressive optical flow estimation for arbitrary-length videos via progressive next-frame forecasting. In: The Fourteenth International Conference on Learning Representations 3
27. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11976–11986 (2022) 8
28. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017) 8
29. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI'81: 7th international joint conference on Artificial intelligence. vol. 2, pp. 674–679 (1981) 2, 3
30. Luo, A., Li, X., Yang, F., Liu, J., Fan, H., Liu, S.: Flowdiffuser: Advancing optical flow estimation with diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19167–19176 (2024) 3, 7, 12
31. Ma, Z., Teed, Z., Deng, J.: Multiview stereo with cascaded epipolar raft. In: European Conference on Computer Vision. pp. 734–750. Springer (2022) 1
32. Mayer, N., Ilg, E., Häusser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) (2016), `http://lmb.informatik.uni-freiburg.de/Publications/2016/MIFDB16`, arXiv:1512.02134 7
33. Mehl, L., Schmalfuss, J., Jahedi, A., Nalivayko, Y., Bruhn, A.: Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In: CVPR. pp. 4981–4991 (2023) 2, 7

34. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: CVPR. pp. 3061–3070 (2015) 1, 2
35. Morimitsu, H., Zhu, X., Ji, X., Yin, X.C.: Recurrent partial kernel network for efficient optical flow estimation. In: AAAI. vol. 38, pp. 4278–4286 (2024) 7, 9, 12
36. Ngo, T.D., Zhuang, P., Gan, C., Kalogerakis, E., Tulyakov, S., Lee, H.Y., Wang, C.: Delta: Dense efficient long-range 3d tracking for any video. arXiv preprint arXiv:2410.24211 (2024) 3
37. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.Y., Xu, H., Sharma, V., Li, S.W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: Dinov2: Learning robust visual features without supervision (2023) 3, 4, 8
38. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. ArXiv (2019) 8
39. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 12179–12188 (2021) 4
40. Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network. In: CVPR. pp. 4161–4170 (2017) 2, 3
41. Richter, S.R., Hayder, Z., Koltun, V.: Playing for benchmarks. In: ICCV. pp. 2213–2222 (2017) 7
42. Sand, P., Teller, S.: Particle video: Long-range motion estimation using point trajectories. International journal of computer vision **80**(1), 72–91 (2008) 3
43. Saxena, S., Herrmann, C., Hur, J., Kar, A., Norouzi, M., Sun, D., Fleet, D.J.: The surprising effectiveness of diffusion models for optical flow and monocular depth estimation **36**, 39443–39469 (2023) 3, 12
44. Shah, J., Bikshandi, G., Zhang, Y., Thakkar, V., Ramani, P., Dao, T.: Flashattention-3: Fast and accurate attention with asynchrony and low-precision. Advances in Neural Information Processing Systems **37**, 68658–68685 (2024) 8
45. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1874–1883 (2016) 4
46. Shi, X., Huang, Z., Bian, W., Li, D., Zhang, M., Cheung, K.C., See, S., Qin, H., Dai, J., Li, H.: Videoflow: Exploiting temporal cues for multi-frame optical flow estimation. In: ICCV. pp. 12469–12480 (2023) 2, 3, 7, 12
47. Shi, X., Huang, Z., Li, D., Zhang, M., Cheung, K.C., See, S., Qin, H., Dai, J., Li, H.: Flowformer++: Masked cost volume autoencoding for pretraining optical flow estimation. In: CVPR. pp. 1599–1610 (2023) 2, 3, 7
48. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., Massa, F., Haziza, D., Wehrstedt, L., Wang, J., Darcet, T., Moutakanni, T., Sentana, L., Roberts, C., Vedaldi, A., Tolan, J., Brandt, J., Couprie, C., Mairal, J., Jégou, H., Labatut, P., Bojanowski, P.: DINOv3 (2025), `https://arxiv.org/abs/2508.10104` 3, 4
49. Sun, D., Roth, S., Black, M.J.: Secrets of optical flow estimation and their principles. In: 2010 IEEE computer society conference on computer vision and pattern recognition. pp. 2432–2439. IEEE (2010) 2

50. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: CVPR. pp. 8934–8943 (2018) 2, 3, 7, 9, 12
51. Sun, S., Liu, J., Li, H., Liu, G., Li, T., Gao, W.: Streamflow: streamlined multi-frame optical flow estimation for video sequences **37**, 9205–9228 (2025) 2, 3, 7, 9, 12
52. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: European conference on computer vision. pp. 402–419. Springer (2020) 2, 3, 6, 7, 9, 11, 12
53. Truong, P., Danelljan, M., Timofte, R.: Glu-net: Global-local universal network for dense flow and correspondences. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6258–6268 (2020) 2
54. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2025) 2, 3, 4, 7, 8
55. Wang, W., Zhu, D., Wang, X., Hu, Y., Qiu, Y., Wang, C., Hu, Y., Kapoor, A., Scherer, S.: Tartanair: A dataset to push the limits of visual slam (2020) 7
56. Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., He, T.: π^3: Scalable permutation-equivariant visual geometry learning (2025), `https://arxiv.org/abs/2507.13347` 2
57. Wang, Y., Deng, J.: Waft: Warping-alone field transforms for optical flow. arXiv preprint arXiv:2506.21526 (2025) 2, 3, 7, 8, 9, 10, 11, 12
58. Wang, Y., Lipson, L., Deng, J.: Sea-raft: Simple, efficient, accurate raft for optical flow. arXiv preprint arXiv:2405.14793 (2024) 2, 3, 7, 8, 9, 10, 11, 12
59. Weinzaepfel, P., Lucas, T., Leroy, V., Cabon, Y., Arora, V., Brégier, R., Csurka, G., Antsfeld, L., Chidlovskii, B., Revaud, J.: Croco v2: Improved cross-view completion pre-training for stereo matching and optical flow. In: ICCV. pp. 17969–17980 (2023) 1, 2, 3, 9, 12
60. Wen, B., Trepte, M., Aribido, J., Kautz, J., Gallo, O., Birchfield, S.: Foundationstereo: Zero-shot stereo matching. CVPR (2025) 3
61. Wu, G., Liu, X., Luo, K., Liu, X., Zheng, Q., Liu, S., Jiang, X., Zhai, G., Wang, W.: Accflow: Backward accumulation for long-range optical flow. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12119–12128 (2023) 3, 9, 11
62. Xu, G., Chen, S., Jia, H., Feng, M., Yang, X.: Memory-efficient optical flow via radius-distribution orthogonal cost volume. arXiv preprint arXiv:2312.03790 (2023) 3
63. Xu, H., Zhang, J., Cai, J., Rezatofighi, H., Tao, D.: Gmflow: Learning optical flow via global matching. In: CVPR. pp. 8121–8130 (2022) 2, 3, 5, 7
64. Xu, H., Zhang, J., Cai, J., Rezatofighi, H., Yu, F., Tao, D., Geiger, A.: Unifying flow, stereo and depth estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023) 2, 3, 5, 7
65. Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H.: Depth anything v2. arXiv:2406.09414 (2024) 3
66. Zhang, Y., Keetha, N., Lyu, C., Jhamb, B., Chen, Y., Qiu, Y., Karhade, J., Jha, S., Hu, Y., Ramanan, D., et al.: Ufm: A simple path towards unified dense correspondence with flow. arXiv preprint arXiv:2506.09278 (2025) 2, 3, 7, 8, 9, 10
67. Zhao, S., Zhao, L., Zhang, Z., Zhou, E., Metaxas, D.: Global matching with overlapping attention for optical flow estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17592–17601 (2022) 2
68. Zheng, Y., Harley, A.W., Shen, B., Wetzstein, G., Guibas, L.J.: Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In: ICCV (2023) 3, 9, 11

69. Zheng, Z., Nie, N., Ling, Z., Xiong, P., Liu, J., Wang, H., Li, J.: Dip: Deep inverse patchmatch for high-resolution optical flow. In: CVPR. pp. 8925–8934 (2022) 3, 7
70. Zhou, S., He, R., Tan, W., Yan, B.: Samflow: Eliminating any fragmentation in optical flow with segment anything model. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 7695–7703 (2024) 3, 7, 8, 10, 12