FreeFlow: A Bias-free Hierarchical Transformer for Optical Flow Estimation

Vladislav Bargatin[1,2*], Alexander Yakovenko[1,2,3*†], Khaled Abud[1,2,3], and Dmitriy Vatolin[3]

[1] AI Center, Lomonosov MSU, Moscow, Russia
[2] Lomonosov Moscow State University, Moscow, Russia
[3] MSU Institute for Artificial Intelligence, Moscow, Russia
{vladislav.bargatin, alexander.yakovenko}@graphics.cs.msu.ru,
{khaled.abud, dmitriy}@graphics.cs.msu.ru
https://github.com/msu-video-group/freeflow

Abstract. Optical flow methods typically rely on task-specific inductive biases, such as correlation volumes, feature warping, and iterative refinement, among others, to reach high accuracy. While effective, such biases constrain the model to predefined heuristics, which can limit its expressivity and lead to more complex pipelines and additional computational cost. We present FreeFlow, a hierarchical transformer built without any flow-specific components, using instead a single feed-forward encoder–decoder. FreeFlow combines three attention variants: window attention for local processing, shifted-window attention for cross-window information exchange, and a global attention operating at a reduced resolution. The resulting architecture scales naturally with model capacity, enabling a consistent accuracy gain from small to large variants. Despite the absence of standard inductive biases, FreeFlow achieves state-of-the-art results on major benchmarks, including Sintel (0.68/1.48 EPE on Clean/Final), KITTI-2015 (3.23 Fl-all), and Spring (3.192 1px), while remaining memory efficient at 1080p inference.

Keywords: Optical flow · Vision Transformers · High-resolution

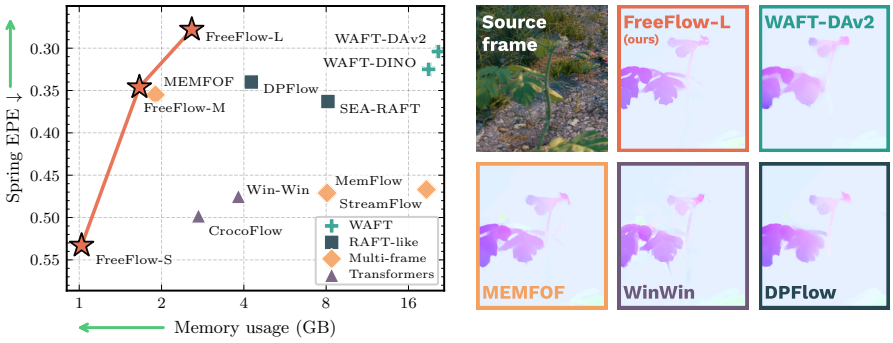
1 Introduction

Optical flow estimation (the dense per-pixel motion between frames) is a fundamental task in low-level vision, with applications ranging from video understanding [32,44,62] and tracking to video restoration and synthesis [6,12,24,57].

Early optical flow methods posed estimation as variational optimization [10, 27], later accumulating stronger regularization [41], coarse-to-fine schemes, and hand-crafted descriptors [53] at the cost of growing complexity.

Deep learning initially simplified optical flow, as FlowNet [9] showed that a feed-forward network can regress flow directly, but later work reintroduced

* Equal contribution
† Corresponding author



(a) Spring EPE (↓) versus 1080p inference memory (GB), with FreeFlow variants (S/M/L) illustrating model scaling.

(b) Qualitative comparison of predicted flow fields on a crop of a visually challenging scene with heavy blur (best viewed zoomed in).

Fig. 1: FreeFlow overview. Our method achieves state-of-the-art accuracy on Spring with low 1080p inference memory and the sharpest motion borders among strong baselines, as evidenced by (a) the accuracy–memory scatter plot and (b) the qualitative crop gallery.

classical biases in highly effective yet hard-wired architectures. PWC-Net [42] combined pyramids, warping, and local correlation volumes, while RAFT [45] popularized iterative refinement over an all-pairs correlation volume and convex upsampling; many follow-ups then build upon these foundations with additional modules for occlusions [14], temporal cues [7, 38], and training/inference refinements [50]. Which improve accuracy, but lead to complex pipelines that are harder to modify, scale, and repurpose beyond optical flow.

In parallel, the broader computer vision literature has moved in the opposite direction: vision transformers [8] increasingly replace bespoke pipelines in detection [5], segmentation [16, 18], depth prediction [58, 59], and 3D tasks [15, 46, 47], driven by the observation that generic, data-driven feed-forward models can learn the required structure from scale and supervision. This trend motivates revisiting optical flow through the same lens: *can we omit flow-specific components and still obtain a state-of-the-art approach?*

Several recent approaches move toward more generic architectures, but still retain flow-specific structure or incur practical limitations. DDVM [37] casts optical flow prediction in a diffusion framework, but the prediction is still produced through iterative process. CroCo [52] is close to a pure transformer, yet it operates at a relatively small fixed resolution and typically requires tiling for higher-resolution inputs, while still relying on a large convolutional decoder. Most recently, WAFT [49] and GeoViT [54] explicitly aim for generality, but still rely on iterations and require warping (of features and input frames, respectively). Additionally, these approaches are trained at relatively small, sub-megapixel resolutions, which might become a limiting factor for accuracy at high-resolution inference [1].

We therefore propose FreeFlow, a bias-free hierarchical transformer for optical flow estimation. FreeFlow is designed for high-resolution processing, which recent work has shown to be beneficial for optical flow [1] and depth estimation [3]. At high resolution, accurate flow requires both strong local reasoning (to preserve fine structures and motion boundaries) and effective long-range information flow (to resolve large displacements). FreeFlow addresses this with a hierarchical attention design that combines tiled processing with repeated local–global feature interaction: local attention focuses on within-region detail, cross-region exchange propagates information between neighboring tiles, and global mixing enables long-range correspondence. FreeFlow sets a new state-of-the-art on Sintel [4] (EPE clean: 0.68; EPE final: 1.48), KITTI-2015 [30] (Fl-all: 3.23), and Spring [29] (1px: 3.192).

Our key contributions are:

- **Bias-free optical flow transformer.** We introduce FreeFlow, a hierarchical transformer for optical flow built *without common flow-specific inductive biases and modules* (e.g., explicit cost/correlation volumes, warping-based update pipelines, and specialized upsampling), using a single end-to-end trainable architecture.
- **Hierarchical local–global feature interaction for high-resolution processing.** We propose a tiled transformer design with repeated local and long-range feature interaction, enabling accurate flow at high resolutions by combining local detail modeling with global information flow.
- **State-of-the-art performance across benchmarks.** FreeFlow achieves state-of-the-art results on Sintel, KITTI-2015, and Spring, all while being memory-efficient and scalable to smaller parameter counts.

2 Related Works

2.1 Inductive Biases of Optical Flow

Early optical flow methods formulated estimation as variational optimization under photometric constancy and smoothness regularization [10,27,41,53]. Learning-based approaches later treated optical flow as supervised dense prediction [9], and subsequent advances largely introduced explicit architectural priors tailored to correspondence estimation. Representative flow-specific inductive biases include multi-scale pyramids [31, 42], warping-based alignment [42, 49, 54], explicit correlation volumes for large-displacement matching [1, 45], iterative refinement, and convex upsampling [1, 45, 50]. While these choices are effective, they have contributed to increasingly complex pipelines; recent work has started to remove individual priors [17] and move toward more generic formulations [37], but existing methods typically retain some of these biases rather than eliminating them entirely.

2.2 Vision Transformers in Dense Prediction

Vision transformers [8] (ViTs) are now widely used for dense prediction and geometry-oriented tasks, enabled by large-scale data and a general architecture

Table 1: Architectural Inductive Biases in Optical Flow Methods. We summarize common design choices used to inject task-specific structure compared to our approach, which has no flow-specific inductive biases and uses a simple decoder head. "DPT head" refers to the decoder head from Dense Prediction Transformer (DPT) [33].

Method	Correlation Volume	Feature Warping	Pyramid Refinement	Iterative Refinement	Convex Upsampling	Inference time Tiling	Flow Head
PWC-Net [42]	✓	✓	✓	×	×	×	×
RAFT [45]	✓	×	×	✓	✓	×	×
FlowFormer [11]	✓	×	×	✓	✓	✓	×
UniMatch [56]	×	✓	✓	✓	✓	×	×
TransFlow [26]	✓	×	×	✓	✓	×	×
DPFlow [31]	✓	×	✓	✓	✓	×	×
MEMFOF [1]	✓	×	×	✓	✓	×	×
WAFT [49]	×	✓	×	✓	✓	×	×
Geo-VIT [54]	×	✓	×	✓	✓	✓	×
CroCo-Flow [52]	×	×	×	×	×	✓	DPT head
Win-Win [20]	×	×	×	×	×	×	DPT head
FreeFlow (ours)	×	×	×	×	×	×	Simple Conv.

that can learn the dependencies from supervision rather than relying on handcrafted priors. This has led to strong results across depth [3, 33, 58, 59], segmentation [16, 18], and 3D settings [15, 46, 47], where feed-forward transformer backbones provide a common foundation for dense outputs and geometric reasoning. Importantly, many of these systems remain architecturally simple. A common pattern is a ViT backbone combined with a convolutional prediction head or decoder for producing dense maps [33,34]. Other approaches further reduce decoder structure and rely on minimal output token processing, suggesting that heavy convolutional decoders are not necessary to obtain competitive dense predictions [15, 16]. However, scaling such models to high resolution is challenging; common workarounds such as downsampling or tiling can harm accuracy and limit long-range interaction. Swin [22, 25] addresses the issue by using local-window self-attention and shifted windows to pass information across window boundaries, while Hiera [36] provides a simple hierarchical multi-scale ViT backbone for large images. Alternatively, DepthPro does high-resolution processing via a multi-scale pyramid design with late feature fusion [3].

2.3 Transformers in Optical Flow Estimation

Recent transformer-based optical flow methods differ mainly in which flow-specific inductive biases they retain. Some methods keep explicit matching as a central operation via correlation/cost-based similarity: UniMatch and TransFlow follow this direction [26, 56], while FlowFormer explicitly constructs and processes a 4D cost volume with a transformer-style decoder [11].

While others move closer to generic transformer formulations, they unfortunately still leave some biases in place. CroCo-Flow was among the first transformer approaches with competitive accuracy, leveraging binocular pretraining for dense matching, but it relies on slow dense tiling at high resolution [52]. Win-

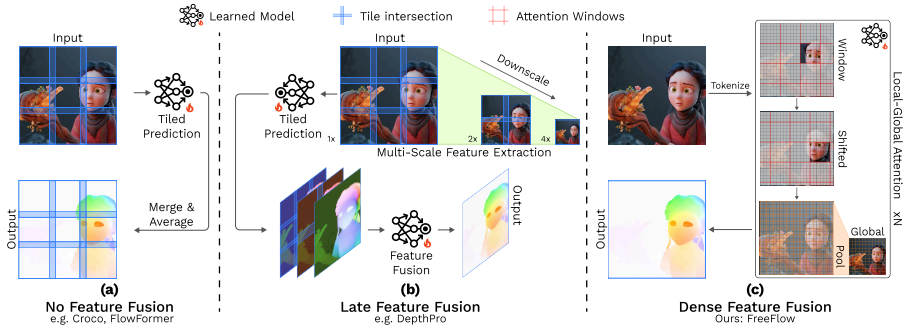


Fig. 2: Comparison of high-resolution prediction techniques. (a) *No Feature Fusion:* per-tile prediction without inter-tile feature exchange (limited global coherence; long-range motions spanning multiple tiles not captured). (b) *Late Feature Fusion:* multi-scale features are fused only in the decoder (global context arrives late and may not propagate to full-resolution features; with small overlap and no information flow across tile borders, seams may remain visible; see supplementary). (c) *Dense Feature Fusion (ours):* repeated cross-tile/global feature exchange across the network (information flow is not restricted to the decoder).

Win builds on CroCo to enable FullHD training and inference without tiling, but does not improve over CroCo-Flow in accuracy [20]. WAFT and GeoViT remove cost volumes but retain iterative warping-based updates (warping features and input images, respectively) [49, 54]. Overall, transformers have often been incorporated by incrementally replacing parts of established optical flow pipelines, which can improve performance yet further diversify and complicate the set of design choices. We summarize these choices in Tab. 1 using common inductive biases and compare to our bias-free approach.

3 Method

In this section, we present FreeFlow, an inductive-bias-free transformer for optical flow estimation (Fig. 3). We design this approach with two goals in mind. First, we aim to demonstrate that state-of-the-art optical flow can be achieved without the flow-specific architectural modules that dominate modern pipelines, such as correlation volumes, feature warping, iterative refinement, etc. (summarized in Tab. 1). Second, we seek a design that remains practical at high resolutions and ensures repeated information exchange across processing scales.

To this end, we study existing high-resolution prediction strategies, identify their limitations, and derive a principled alternative (Fig. 2). A straightforward solution is inference-time tiling in CroCo-/FlowFormer-style pipelines (Fig. 2a), but tiles interact only through late-stage averaging, so information does not flow across tile borders during feature extraction, and the number of forward passes grows with resolution. An alternative is multi-scale decoding with *Late Feature Fusion* as in DepthPro-like designs (Fig. 2b), which reduces the number

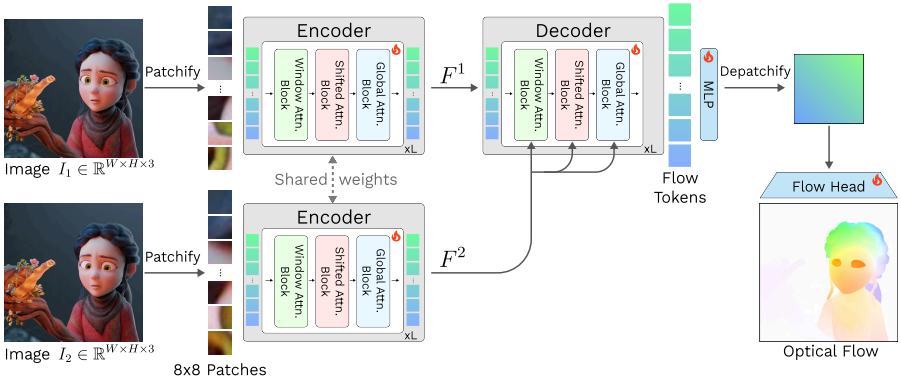


Fig. 3: Method overview. Given an image pair (I_1, I_2), we patchify each input into 8×8 tokens and extract features using two shared-weight encoders composed of Window, Shifted-Window, and Global attention blocks (Fig. 4), producing F^1 and F^2. A transformer decoder combines self-attention with cross-attention between F^1 and F^2 to form flow tokens, which are depatchified and mapped to a dense optical flow field by a lightweight prediction head.

of forward passes, but keeps information exchange delayed and tied to fixed-resolution assumptions, and is unreliable in the binocular setting when objects cross tile boundaries. FreeFlow resolves these issues by using a fixed and efficient tiling scheme while enabling dense feature exchange throughout the network (Fig. 2c), combining local processing with cross-tile and global interactions.

We next give a functional description of the architecture and then formalize the attention variants used by FreeFlow.

3.1 Approach

We adopt the CroCo/DUSt3R/MASt3R encoder–decoder high-level architecture for binocular reasoning, i.e., a Siamese ViT encoder followed by a decoder that alternates self- and cross-attention between the two views.

Given two input images $I_1, I_2 \in \mathbb{R}^{H\times W\times 3}$, we embed each image into a sequence of non-overlapping $P\times P$ patches (with P=8) using a standard patch projection, producing token sequences $X_1, X_2 \in \mathbb{R}^{N\times D}$ where $N = \frac{H}{P} \cdot \frac{W}{P}$. Both sequences are processed by a Siamese transformer encoder with shared weights to obtain feature representations

$$F^1 = \text{Encoder}(X_1), \qquad F^2 = \text{Encoder}(X_2), \tag{1}$$

with $F^1, F^2 \in \mathbb{R}^{N\times D}$.

A transformer decoder then produces flow tokens

$$Z = \text{Decoder}(F^1, F^2), \tag{2}$$

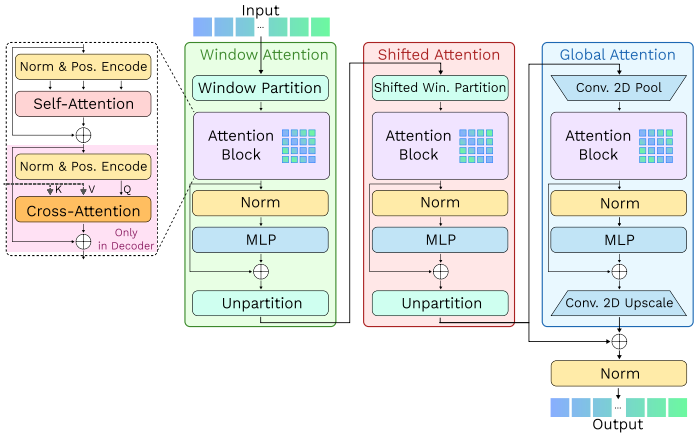


Fig. 4: Attention block variants. Our architecture uses three attention patterns: (i) *Window* attention over a 4×4 partition (16 non-overlapping windows), (ii) *Shifted-Window* attention with a half-window offset to exchange information across window boundaries, and (iii) *Global* attention applied at 2× lower spatial resolution via down/up-sampling. See Fig. 2 for the partitioning visualization.

where each decoder block combines self-attention over the current tokens with cross-attention from F^1 (queries) to F^2 (keys/values), enabling repeated information exchange between the two views. The output $Z \in \mathbb{R}^{N \times D}$ is reshaped into a spatial feature map $Z_{\text{map}} \in \mathbb{R}^{\frac{H}{P} \times \frac{W}{P} \times D}$ and mapped to a dense flow and confidence field $U \in \mathbb{R}^{H \times W \times 5}$ by a prediction head.

Attention Block Types. To predict highly-detailed globally consistent flow fields, FreeFlow uses a hierarchical attention design that mixes local processing, cross-window exchange, and global context. Each block follows the CroCo-style transformer structure: encoder blocks apply self-attention, while decoder blocks additionally use cross-attention to utilize tokens from the other view. We use three attention variants (Fig. 4):

- *Window (Win) Attention Block.* The token map of size $\frac{H}{8} \times \frac{W}{8}$ is partitioned into a 4×4 grid of 16 non-overlapping windows, each containing $\frac{H}{32} \times \frac{W}{32}$ tokens. Attention is computed independently within each window.
- *Shifted-Window (Swin) Attention Block.* To enable information flow across window (and tile) boundaries, we apply a half-window shift by $\lfloor \frac{W}{64} \rfloor$ tokens horizontally and $\lfloor \frac{H}{64} \rfloor$ tokens vertically, partition into the same 4×4 windows, perform window attention, and shift back.
- *Global Attention Block.* To incorporate global context at controlled cost, we downsample by 2× with a stride-2 convolution, apply full attention at the reduced resolution, and upsample by 2× with a stride-2 transposed convolution. We apply normalization after the residual connection.

Each encoder/decoder layer applies the blocks in the fixed order: Win, Swin, and Global. Since attention cost scales quadratically with the number of tokens,

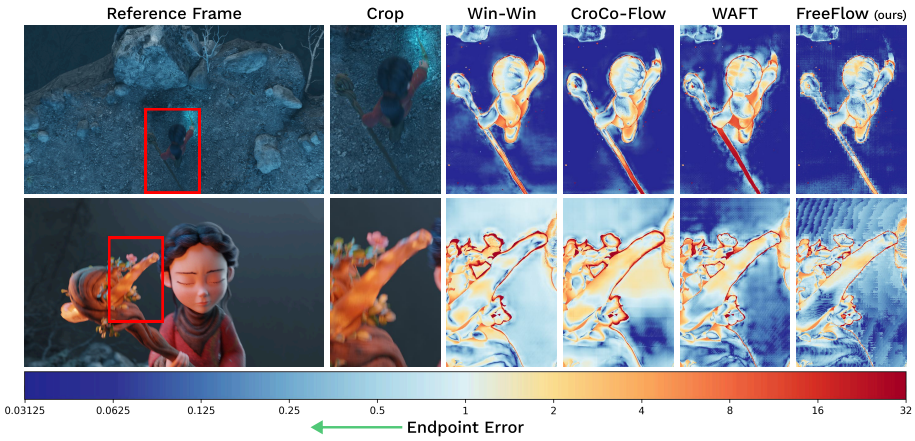


Fig. 5: Qualitative comparison on the Spring benchmark [29]. Examples of error-maps of Win-Win [20], CroCo-Flow [52], WAFT [49], and FreeFlow predictions; the colorbar represents endpoint error. Our approach combines high level of detail (note the hole in the staff that only our method captures and the thin object bounds) with high motion consistency (note the staff's end in the top example and the low error in the background of the bottom example). Crops are sourced from official leaderboard submissions.

the 2× downsampling in the Global block keeps its attention cost comparable to Win/Swin at the original resolution. To encode token positional information within the image, we rely on Rotary Positional Embedding (RoPE) [40].

Attention Scale Factor. Following prior work [7] on resolution-adaptive attention scaling, we multiply the attention logits by a logarithmic factor of the token count, which improves generalization when running inference at resolutions higher than those seen during training. For a token map of size $\frac{H}{8} \times \frac{W}{8}$ we use

$$\text{Self-Attention}(Q, K, V) = \text{Softmax}\left(\frac{\log(H/8 \times W/8)}{\sqrt{D}} \times QK^T\right) \times V \quad (3)$$

$$\text{Cross-Attention}(Q_1, K_2, V_2) = \text{Softmax}\left(\frac{\log(H/8 \times W/8)}{\sqrt{D}} \times Q_1 K_2^T\right) \times V_2 \quad (4)$$

Unlike prior formulations that normalize the factor to be 1 at the training token count, we use the unnormalized variant and found it to work well in practice.

Flow Head. Most high-performing optical flow pipelines rely on flow-specific prediction machinery, such as iterative update stages and convex upsampling. In addition, bias-free transformer baselines often use DPT-style heads from [33] that aggregate features from multiple layers (and, in CroCo-style designs, may also reuse encoder features) to form the final prediction. In contrast, FreeFlow

Table 2: Training procedure details. Dataset abbreviations: TA: TartanAir [48], T: Things [28], S: Sintel [4], K: KITTI-2015 [30], H: HD1K [19]. Inspired by SEA-RAFT [50] and MEMFOF [1], the dataset distributions for the TaTSKH stages are TA (0.23), S(0.25), T(0.24), K(0.09), H(0.19).

Stage	Weights	Datasets	Scale	Crop size	LR	WD	Batch	Steps
Pretrain	—	ARKitScenes [2] MegaDepth [21] 3DStreetView [60]	1x	[224, 224]	8e-4	5e-2	2048	346k
TaTSKH	Pretrain	TA+T+S+K+H	2x	10880 tok.	4e-5	1e-2	32	450k
TaTSKH-hq	TaTSKH	TA+T+S+K+H	2x	32640 tok.	1e-5	1e-5	32	90k
Sintel-ft	TaTSKH-hq	S	2x	[872, 2048]	1e-5	1e-5	32	12.5k
KITTI-ft	TaTSKH-hq	K	2x	[750, 2484]	1e-5	1e-5	32	2.5k
Spring-ft	TaTSKH-hq	Spring [29]	1x	[1080, 1920]	1e-5	1e-5	32	60k

predicts flow directly from the final decoded patch map using a simple three-layer head, without iterative refinement or convex upsampling.

Given $F \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times D}$, we apply a 3×3 convolution to expand channels to $4D$, followed by a 1×1 convolution to 4096 channels, and a transposed convolution with kernel and stride 8 to upsample to $H \times W$. The head outputs $\mathbb{R}^{H \times W \times 5}$, where the first two channels represent optical flow and the remaining three parameterize the uncertainty terms used by the mixture-of-Laplace loss (following SEA-RAFT [50]). Finally, we multiply the flow channels by 8 (the patch size) to obtain flow in pixel units.

4 Experiments

We first detail our training pipeline, consisting of cross-view completion pretraining followed by finetuning for optical flow. We evaluate our method on three popular optical flow benchmarks: Spring [29] (high-resolution real-world sequences), Sintel [4] (synthetic scenes with complex motion and rendering effects), and KITTI-2015 [30] (real driving scenes). Finally, we provide ablations of key design choices, including the attention configuration, pretraining masking ratio, and model scaling.

4.1 Training Details

Following CroCo [51, 52], we first pre-train our model on the cross-view completion task and then finetune the resulting weights with a new head for optical flow estimation. In cross-view completion, a large fraction of patches in one view is replaced by a learned token e_{mask}, and the model reconstructs the missing content conditioned on the second view. Due to its two-image nature, this objective encourages learning dense long-range correspondences, which is well aligned with

the downstream task of binocular matching. Training details and datasets are summarized in Tab. 2; please refer to the supplementary for additional details.

Pre-train. We follow the pre-training stage protocol from CroCo with minor adjustments. During cross-view completion training, a model takes as input two images that represent different views of the same scene: one view is severely masked, and the model has to predict the masked regions using the information from the second view. We sample image pairs from ARKitScenes [2], MegaDepth [21], and 3DStreetView [60], resulting in 3.7M data samples in total. We use fixed-size crops of 224×224 resolution and pretrain the model for 346k steps.

The only significant difference from the CroCo setup is when the learned masked patch representation e_{mask} is introduced. In CroCo, masked tokens are removed from the first view and the corresponding e_{mask} tokens are added only at the decoder input. Here, due to the hierarchical nature of our model, we replace masked patches with e_{mask} at the encoder input, while keeping the completion objective unchanged.

Optical Flow Finetune. The finetuning stage protocol is inspired by MEMFOF [1]; specifically, we adopt their 2× upsampling of training frames, which better matches the motion distribution of FullHD inputs and improves high-resolution performance. Unlike MEMFOF and other curriculum-based training recipes that use multiple sequential stages, we use a single main dataset mixture, denoted TaTSKH in Tab. 2, for simplicity. For additional speed, we split this finetuning into a low- and high-token-count stage (TaTSKH and TaTSKH-hq in Tab. 2).

Instead of using a fixed crop size, to avoid unnecessary padding and to expose the model to a wider motion range, we use variable-resolution training with a fixed token budget per minibatch. More specifically, for each sample we randomly choose one spatial dimension (height or width), sample its value, and set the other dimension to the largest value such that the resulting token count does not exceed the prescribed budget. This produces rectangular crops with varying aspect ratios while keeping compute and memory controlled. The implementation is straightforward as we use a batch size of 1 sample/GPU during the finetuning stage. For benchmark submissions, we further finetune with fixed crop sizes (Sintel-ft, KITTI-ft, Spring-ft in Tab. 2). Following SEA-RAFT [50], we use the Mixture-of-Laplace loss.

In total, it takes from 4 to 5 days to pre-train and around 3 days to finetune our largest model on 32 GPUs.

4.2 Results

We adopt four widely used metrics from established benchmarks in this study: endpoint error (EPE), 1-pixel outlier rate (1px), Fl-score, and WAUC error. Please refer to [4, 29–31, 35] or the supplementary for their definitions.

Results on Spring. FreeFlow achieves state-of-the-art performance on Spring. FreeFlow-L sets the best EPE and Fl among all compared methods (Tab. 3), while remaining on par with the strongest approaches in 1px and WAUC; in

Table 3: Spring benchmark results. "*" indicates that it was submitted by the Spring team without being finetuned on the provided training set. Speed (runtime) and peak GPU memory consumption were measured on a Nvidia RTX 3090 GPU (24 GB) with automatic mixed precision and without memory efficient correlation volumes, where "—" denotes no available data or implementation for a given model and "†" denotes our best estimate based on the authors description. "MF" indicates that the method uses multiple frames (three or more). The best results are indicated in **bold**, second-best are underlined, third best are indicated in *italic*. Method configurations are taken from submissions to the Spring benchmark if present, and from submissions to the Sintel benchmark otherwise.

Method	Inf. Cost (1080p)		Params (M)	Spring			
	Memory (GB)	Time (ms)		1px ↓	EPE ↓	Fl ↓	WAUC ↑
FlowNet2 [13]	3.01	110	162.52	6.710*	1.040*	2.823*	90.907*
PWC-Net [42]	0.57	48	9.37	82.265*	2.288*	4.889*	45.670*
RAFT [45]	7.97	406	5.26	6.790*	1.476*	3.198*	90.920*
GMA [14]	11.81	830	5.88	7.074*	0.914*	3.079*	90.722*
GMFlow [55]	8.22	8024	4.72	10.355*	0.945*	2.952*	82.337*
FlowFormer [11]	1.90	2084†	16.17	6.510*	0.723*	2.384*	91.679*
SEA-RAFT (M) [50]	8.12	198	19.67	3.686	0.363	1.347	94.534
DPFlow [31]	4.26	401	10.02	3.442	0.340	1.311	94.980
MemFlow$^{(MF)}$ [7]	8.06	754	6.27	4.482	0.471	1.416	93.855
StreamFlow$^{(MF)}$ [43]	18.61	898	14.25	4.152	0.467	1.424	94.404
MEMFOF$^{(MF)}$ [1]	1.90	262	75.78	3.289	0.355	1.238	*95.186*
ARFlow$^{(MF)}$ [23]	—	—	76.50	*3.265*	0.353	1.212	**95.283**
CroCo-Flow [52]	2.73	3266	447.47	4.565	0.498	1.508	93.660
Win-Win [20]	3.82†	305†	229.77†	5.371	0.475	1.621	92.720
WAFT-DAv2-a2 [49]	20.58	489	56.93	3.298	0.304	*1.197*	94.990
WAFT-DINOv3-a2 [49]	18.96	408	56.47	**3.182**	*0.325*	1.246	95.051
FreeFlow-S (ours)	1.02	144	34.58	5.087	0.533	1.452	90.196
FreeFlow-M (ours)	1.66	325	102.46	3.392	0.346	1.171	94.919
FreeFlow-L (ours)	2.58	607	230.72	3.192	**0.278**	**1.048**	95.235

particular, it attains the best WAUC among two-frame methods. Compared to WAFT-DAv2-a2, FreeFlow-L improves EPE by 9% and reduces Fl by 14%. Owing to tiling-free native 1080p inference, FreeFlow preserves fine detail while maintaining global motion consistency (Fig. 5). We further highlight that native 1080p processing is possible within a low inference memory budget (Fig. 1).

Results on Sintel and KITTI. Following MEMFOF, we finetune on 2× upsampled frames. Accordingly, for Sintel and KITTI submissions we upscale input images by 2× and downscale the predicted flow by 2×. FreeFlow-L ranks first on Sintel on both Clean and Final (Tab. 4), improving over GeoVIT [54] by 14% on Clean (0.79→0.68) and 10% over VideoFlow-MOF [38] on Final (1.65→1.48). Notably, FreeFlow-M is already highly competitive: it is second only to FreeFlow-L on Clean, and ranks fourth on Final, surpassed only by the 3- and 5-frame VideoFlow variants. On KITTI-2015, FreeFlow-L achieves 3.23 Fl-all, outperforming all non-stereo and non-multiframe methods on KITTI-15 (Tab. 4). Qualitative

Table 4: Sintel [4] and KITTI-15 [30] benchmark results. Sintel uses EPE as it's metric for both splits, while KITTI-15 uses the Fl-all outliers metric. "—" indicates no published results and "MF" indicates that the method used multiple frames (three or more) to generate it's submissions.

Method	Sintel		KITTI-15
	Clean ↓	Final ↓	Fl-all ↓
FlowNet2 [13]	4.16	5.74	10.41
PWC-Net [42]	3.90	5.04	9.60
RAFT [45]	1.61	2.86	5.10
GMA [14]	1.39	2.47	5.15
GMFlow+ [56]	1.03	2.37	4.49
FlowFormer [11]	1.16	2.09	4.68
FlowFormer++ [39]	1.07	1.94	4.52
TransFlow [26]	1.06	2.08	4.32
SEA-RAFT (L) [50]	1.31	2.60	4.30
DPFlow [31]	1.05	1.98	3.56
VideoFlow-BOF$^{(MF)}$ [38]	1.00	*1.71*	4.44
VideoFlow-MOF$^{(MF)}$ [38]	0.99	1.65	3.65
MEMFOF$^{(MF)}$ [1]	0.99	1.94	2.94
MEMFOF-XL$^{(MF)}$ [1]	0.93	1.89	—
ARFlow$^{(MF)}$ [23]	0.96	1.79	**2.85**
DDVM [37]	1.75	2.48	3.26
CroCo-Flow [52]	1.09	2.44	3.64
Win-Win [20]	1.15	2.34	—
WAFT-DAv2-a2 [49]	0.94	2.33	3.31
WAFT-DINOv3-a2 [49]	0.95	2.02	3.56
GeoVIT [54]	0.79	1.88	3.79
FreeFlow-S (ours)	1.03	1.99	4.06
FreeFlow-M (ours)	*0.80*	1.77	3.33
FreeFlow-L (ours)	**0.68**	**1.48**	*3.23*

Fig. 6: Qualitative comparison on Sintel. Examples of FreeFlow-L, GeoViT [54], WAFT-DAv2-a2 [49], CroCo-Flow [52], Win-Win [20], and MEMFOF-XL [1] outputs on the Sintel [4] benchmark. Sourced from the official leaderboard submissions.

results show that the model captures complex motion patterns using only two frames (Fig. 6). Additional visual comparisons and zero-shot evaluations are provided in the supplementary material.

4.3 Ablation Study

Unless stated otherwise, all ablations use a scaled-down FreeFlow configuration with 8 encoder and 8 decoder layers, width 256, and 8 attention heads, and are trained with the same training recipe. Following SEA-RAFT and WAFT, we report results on the Spring sub-validation split (scenes 0045 and 0047) after finetuning on the remaining training data. Additional experiments in the supplementary isolate the architecture from the training procedure and study the effect of adding iterative flow-specific biases back into FreeFlow.

Table 5: Masking ratio and attention-block ablation. Spring sub-validation results for FreeFlow variants with different cross-view completion masking ratios and active attention subblocks (Win/Swin/Global). Gray marks the configuration selected for the main model; best and second-best results are shown in **bold** and underlined, respectively.

Mask. Ratio	Subblock type			Spring (sub-val)	
	Win	Swin	Global	1px ↓	EPE ↓
0.9	×	×	✓	1.133	0.229
0.9	✓	×	✓	0.801	0.190
0.9	✓	✓	×	0.659	0.167
0.9	✓	✓	✓	0.688	0.170
0.925	✓	✓	✓	0.685	0.166
0.95	✓	✓	×	0.658	0.166
0.95	✓	✓	✓	**0.624**	**0.157**
0.975	✓	✓	✓	0.656	0.174

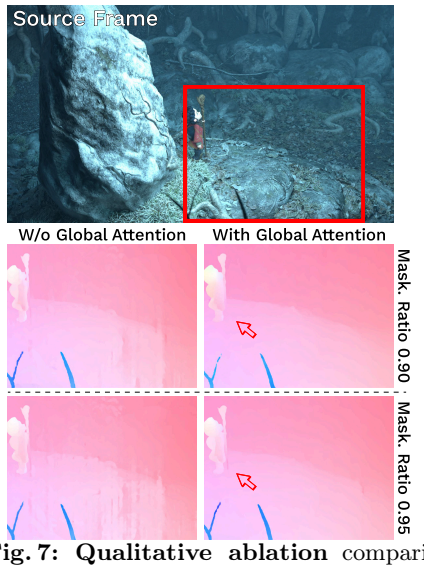


Fig. 7: Qualitative ablation comparison of FreeFlow models with and without Global Attention block and pretrained with different masking ratios on the Spring benchmark. Zoom in for better view.

Architecture and Masking Ratio Ablation. We study the interaction between the cross-view completion masking ratio and the hierarchical attention design of FreeFlow (Tab. 5). The masking ratio controls the fraction of patches in the target view that are replaced by the learned token e_{mask} during pretraining, while the attention configuration determines which subblock types (Win/Swin/Global) are present in the encoder and decoder.

With all three subblocks enabled, a masking ratio of 0.95 performs best, improving over the CroCo default 0.9 by 9.3% in 1px and 7.6% in EPE. This is consistent with FreeFlow using smaller patches than CroCo (8×8 vs. 16×16), which reduces the distance to visible regions and makes a higher mask rate beneficial as it makes the pretraining task sufficiently challenging.

We also observe an interaction between masking ratio and attention configuration. At the CroCo default ratio (0.9), removing Global does not hurt and can even slightly improve the sub-validation metrics, whereas at 0.95 Global becomes important for the best performance. This indicates that the masking ratio is an important hyperparameter that should be chosen jointly with the model architecture. Shifted-Window attention consistently contributes to accuracy, and removing it leads to a clear drop on this split. Additionally, for both masking ratios, qualitative comparisons (Fig. 7) show that removing the Global block can introduce obvious motion inconsistencies, even when the sub-validation metrics change only marginally.

Table 6: Model configurations. Architectural hyperparameters of FreeFlow variants and our *best-effort estimates* for transformer-based baselines. "+" denotes separate encoder and decoder widths (or counts), while "×" denotes repeated recurrent decoder calls.

Name	Patch Sizes	Width	Attn. Count	Attn. Heads	MLP Count	Params (M)
GeoViT	16	1024	24×6	16	24×6	377
WAFT	16	384+384	12+12×5	6+6	12+12×5	56
CroCo-Flow	16	1024+768	24+24	16+12	24+12	447
Win-Win	16	768+768	12+24	12	12+12	230
GMFlow+	8/4	0+128	0+12×2	0+1	0+6×2	4.7
FreeFlow-S	8	256+256	12+24	4+4	12+12	35
FreeFlow-M	8	384+384	18+36	6+6	18+18	102
FreeFlow-L	8	512+512	24+48	8+8	24+24	231

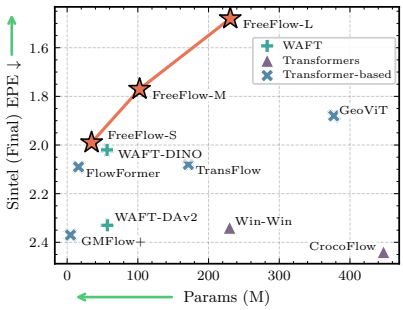


Fig. 8: Model scaling on Sintel. Sintel Final EPE (↓) versus parameter count (M), $\times 10^6$ for FreeFlow variants and transformer-based baselines.

Model Scaling. A benefit of FreeFlow's simple, uniform encoder–decoder design is that it can be scaled in a straightforward manner by adjusting depth and width. We therefore study how performance evolves across model sizes (S/M/L) on common optical flow benchmarks. We follow common ViT scaling [61] best practices by co-scaling depth, width, and the number of attention heads while keeping the overall architecture and patch size fixed. The resulting configurations are summarized in Tab. 6.

As shown in Fig. 8, FreeFlow scales predictably: larger variants yield consistent accuracy gains, while smaller variants retain strong performance, indicating that the approach remains effective even when scaled down. Memory scaling with model size is reported in Fig. 1a.

5 Conclusion

In this work, we introduced FreeFlow, a bias-free hierarchical transformer for optical flow estimation that removes conventional flow-specific design components such as correlation volumes, feature warping, and iterative refinement. Instead, FreeFlow relies on a simple feed-forward encoder–decoder architecture that combines window, shifted-window, and reduced-resolution global attention to capture motion across multiple spatial scales. This design provides a flexible and scalable framework that improves consistently with model capacity while maintaining efficient high-resolution inference.

We show that these standard optical-flow inductive biases are not required to achieve top performance. FreeFlow reaches state-of-the-art results on all popular benchmarks, including Sintel, KITTI-2015, and Spring, demonstrating that strong motion estimation can be obtained from a general-purpose transformer architecture without specialized flow modules. We hope that this work encourages further exploration of simpler and more general architectures for motion estimation and related dense correspondence tasks.

Acknowledgements

The work of Vladislav Bargatin, Alexander Yakovenko and Khaled Abud was supported by the The Ministry of Economic Development of the Russian Federation in accordance with the subsidy agreement (agreement identifier 000000C313925P4H0002; grant No 139-15-2025-012). The research was carried out using the MSU-270 supercomputer of Lomonosov Moscow State University.

References

1. Bargatin, V., Chistov, E., Yakovenko, A., Vatolin, D.: MEMFOF: High-resolution training for memory-efficient multi-frame optical flow estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 8187–8196 (2025). https://doi.org/10.1109/ICCV51701.2025.00767
2. Baruch, G., Chen, Z., Dehghan, A., Feigin, Y., Fu, P., Gebauer, T., Kurz, D., Dimry, T., Joffe, B., Schwartz, A., Shulman, E.: ARKitScenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. In: Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks. vol. 1 (2021)
3. Bochkovskiy, A., Delaunoy, A., Germain, H., Santos, M., Zhou, Y., Richter, S., Koltun, V.: Depth Pro: Sharp monocular metric depth in less than a second. In: International Conference on Learning Representations. vol. 2025, pp. 75602–75637 (2025)
4. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: Computer Vision – ECCV 2012. pp. 611–625. Springer Berlin Heidelberg, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33783-3_44
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Computer Vision – ECCV 2020. pp. 213–229. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_13
6. Chan, K.C., Wang, X., Yu, K., Dong, C., Loy, C.C.: BasicVSR: The search for essential components in video super-resolution and beyond. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4945–4954 (2021). https://doi.org/10.1109/CVPR46437.2021.00491
7. Dong, Q., Fu, Y.: MemFlow: Optical flow estimation and prediction with memory. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 19068–19078 (2024). https://doi.org/10.1109/CVPR52733.2024.01804
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
9. Dosovitskiy, A., Fischer, P., Ilg, E., Häusser, P., Hazirbas, C., Golkov, V., Smagt, P.v.d., Cremers, D., Brox, T.: FlowNet: Learning optical flow with convolutional networks. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 2758–2766 (2015). https://doi.org/10.1109/ICCV.2015.316
10. Horn, B.K., Schunck, B.G.: Determining optical flow. Artificial Intelligence **17**(1-3), 185–203 (1981). https://doi.org/10.1016/0004-3702(81)90024-2

11. 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: Computer Vision – ECCV 2022. pp. 668–685. Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-19790-1_40
12. Huang, Z., Zhang, T., Heng, W., Shi, B., Zhou, S.: Real-time intermediate flow estimation for video frame interpolation. In: Computer Vision – ECCV 2022. pp. 624–642. Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-19781-9_36
13. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: FlowNet 2.0: Evolution of optical flow estimation with deep networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1647–1655 (2017). https://doi.org/10.1109/CVPR.2017.179
14. Jiang, S., Campbell, D., Lu, Y., Li, H., Hartley, R.: Learning to estimate hidden motions with global motion aggregation. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9752–9761 (2021). https://doi.org/10.1109/ICCV48922.2021.00963
15. Jin, H., Jiang, H., Tan, H., Zhang, K., Bi, S., Zhang, T., Luan, F., Snavely, N., Xu, Z.: LVSM: A large view synthesis model with minimal 3d inductive bias. In: International Conference on Learning Representations. vol. 2025, pp. 60001–60021 (2025)
16. Kerssies, T., Cavagnero, N., Hermans, A., Norouzi, N., Averta, G., Leibe, B., Dubbelman, G., De Geus, D.: Your vit is secretly an image segmentation model. In: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 25303–25313 (2025). https://doi.org/10.1109/CVPR52734.2025.02356
17. Kiefhaber, S., Roth, S., Schaub-Meyer, S.: Removing cost volumes from optical flow estimators. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 79–89 (2025). https://doi.org/10.1109/ICCV51701.2025.00015
18. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3992–4003 (2023). https://doi.org/10.1109/ICCV51070.2023.00371
19. Kondermann, D., Nair, R., Honauer, K., Krispin, K., Andrulis, J., Brock, A., Güssefeld, B., Rahimimoghaddam, M., Hofmann, S., Brenner, C., Jähne, B.: The HCI Benchmark Suite: Stereo and flow ground truth with uncertainties for urban autonomous driving. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 19–28 (2016). https://doi.org/10.1109/CVPRW.2016.10
20. Leroy, V., Revaud, J., Lucas, T., Weinzaepfel, P.: Win-Win: Training high-resolution vision transformers from two windows. In: International Conference on Learning Representations. vol. 2024, pp. 48749–48767 (2024)
21. Li, Z., Snavely, N.: MegaDepth: Learning single-view depth prediction from internet photos. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2041–2050 (2018). https://doi.org/10.1109/CVPR.2018.00218
22. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., Timofte, R.: SwinIR: Image restoration using swin transformer. In: 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW). pp. 1833–1844 (2021). https://doi.org/10.1109/ICCVW54120.2021.00210

23. 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: International Conference on Learning Representations. vol. 2026 (2026)
24. Liu, X., Liu, H., Lin, Y.: Video frame interpolation via optical flow estimation with image inpainting. International Journal of Intelligent Systems **35**(12), 2087–2102 (2020). https://doi.org/10.1002/int.22285
25. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin Transformer: Hierarchical vision transformer using shifted windows. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9992–10002 (2021). https://doi.org/10.1109/ICCV48922.2021.00986
26. Lu, Y., Wang, Q., Ma, S., Geng, T., Chen, Y.V., Chen, H., Liu, D.: TransFlow: Transformer as flow learner. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 18063–18073 (2023). https://doi.org/10.1109/CVPR52729.2023.01732
27. 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. Vancouver, Canada (1981)
28. 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: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4040–4048 (2016). https://doi.org/10.1109/CVPR.2016.438
29. 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: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4981–4991 (2023). https://doi.org/10.1109/CVPR52729.2023.00482
30. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3061–3070 (2015). https://doi.org/10.1109/CVPR.2015.7298925
31. Morimitsu, H., Zhu, X., Cesar, R.M., Ji, X., Yin, X.C.: DPFlow: Adaptive optical flow estimation with a dual-pyramid framework. In: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 17810–17820 (2025). https://doi.org/10.1109/CVPR52734.2025.01659
32. Piergiovanni, A., Ryoo, M.S.: Representation flow for action recognition. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9937–9945 (2019). https://doi.org/10.1109/CVPR.2019.01018
33. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 12159–12168 (2021). https://doi.org/10.1109/ICCV48922.2021.01196
34. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K.V., Carion, N., Wu, C.Y., Girshick, R., Dollar, P., Feichtenhofer, C.: SAM 2: Segment anything in images and videos. In: International Conference on Learning Representations. vol. 2025, pp. 28085–28128 (2025)
35. Richter, S.R., Hayder, Z., Koltun, V.: Playing for benchmarks. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 2232–2241 (2017). https://doi.org/10.1109/ICCV.2017.243

36. Ryali, C., Hu, Y.T., Bolya, D., Wei, C., Fan, H., Huang, P.Y., Aggarwal, V., Chowdhury, A., Poursaeed, O., Hoffman, J., Malik, J., Li, Y., Feichtenhofer, C.: Hiera: A hierarchical vision transformer without the bells-and-whistles. In: Proceedings of the 40th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 202, pp. 29441–29454. PMLR (2023)
37. 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. In: Advances in Neural Information Processing Systems. vol. 36, pp. 39443–39469. Curran Associates, Inc. (2023)
38. 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: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 12435–12446 (2023). https://doi.org/10.1109/ICCV51070.2023.01146
39. 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: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1599–1610 (2023). https://doi.org/10.1109/CVPR52729.2023.00160
40. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: RoFormer: Enhanced transformer with rotary position embedding. Neurocomputing **568**, 127063 (2024). https://doi.org/https://doi.org/10.1016/j.neucom.2023.127063
41. 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 (2010). https://doi.org/10.1109/CVPR.2010.5539939
42. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: PWC-Net: Cnns for optical flow using pyramid, warping, and cost volume. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8934–8943 (2018). https://doi.org/10.1109/CVPR.2018.00931
43. Sun, S., Liu, J., Li, H., Liu, G., Li, T.H., Gao, W.: StreamFlow: Streamlined multi-frame optical flow estimation for video sequences. In: Advances in Neural Information Processing Systems. vol. 37, pp. 9205–9228. Curran Associates, Inc. (2024). https://doi.org/10.52202/079017-0292
44. Sun, S., Kuang, Z., Sheng, L., Ouyang, W., Zhang, W.: Optical flow guided feature: A fast and robust motion representation for video action recognition. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1390–1399 (2018). https://doi.org/10.1109/CVPR.2018.00151
45. Teed, Z., Deng, J.: RAFT: Recurrent all-pairs field transforms for optical flow. In: Computer Vision – ECCV 2020. pp. 402–419. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-58536-5_24
46. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: VGGT: Visual geometry grounded transformer. In: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5294–5306 (2025). https://doi.org/10.1109/CVPR52734.2025.00499
47. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: DUSt3R: Geometric 3d vision made easy. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 20697–20709 (2024). https://doi.org/10.1109/CVPR52733.2024.01956

48. 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. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4909–4916 (2020). https://doi.org/10.1109/IROS45743.2020.9341801
49. Wang, Y., Deng, J.: WAFT: Warping-alone field transforms for optical flow. In: International Conference on Learning Representations. vol. 2026 (2026)
50. Wang, Y., Lipson, L., Deng, J.: SEA-RAFT: Simple, efficient, accurate raft for optical flow. In: Computer Vision – ECCV 2024. pp. 36–54. Springer Nature Switzerland, Cham (2025). https://doi.org/10.1007/978-3-031-72667-5_3
51. Weinzaepfel, P., Leroy, V., Lucas, T., Brégier, R., Cabon, Y., Arora, V., Antsfeld, L., Chidlovskii, B., Csurka, G., Revaud, J.: CroCo: Self-supervised pre-training for 3d vision tasks by cross-view completion. In: Advances in Neural Information Processing Systems. vol. 35, pp. 3502–3516. Curran Associates, Inc. (2022)
52. 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: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 17923–17934 (2023). https://doi.org/10.1109/ICCV51070.2023.01647
53. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: DeepFlow: Large displacement optical flow with deep matching. In: 2013 IEEE International Conference on Computer Vision. pp. 1385–1392 (2013). https://doi.org/10.1109/ICCV.2013.175
54. Wu, H., Cheng, K.T., Lin, S., Wu, Z.: A study of finetuning video transformers for multi-view geometry tasks. Proceedings of the AAAI Conference on Artificial Intelligence (2026). https://doi.org/10.1609/aaai.v40i13.38038
55. Xu, H., Zhang, J., Cai, J., Rezatofighi, H., Tao, D.: GMFlow: Learning optical flow via global matching. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8111–8120 (2022). https://doi.org/10.1109/CVPR52688.2022.00795
56. 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 **45**(11), 13941–13958 (2023). https://doi.org/10.1109/TPAMI.2023.3298645
57. Xu, X., Siyao, L., Sun, W., Yin, Q., Yang, M.H.: Quadratic video interpolation. In: Advances in Neural Information Processing Systems. vol. 32, pp. 1645–1654. Curran Associates, Inc. (2019)
58. Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H.: Depth Anything: Unleashing the power of large-scale unlabeled data. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10371–10381 (2024). https://doi.org/10.1109/CVPR52733.2024.00987
59. Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H.: Depth Anything V2. In: Advances in Neural Information Processing Systems. vol. 37, pp. 21875–21911. Curran Associates, Inc. (2024). https://doi.org/10.52202/079017-0688
60. Zamir, A.R., Wekel, T., Agrawal, P., Wei, C., Malik, J., Savarese, S.: Generic 3d representation via pose estimation and matching. In: Computer Vision – ECCV 2016. pp. 535–553. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_33
61. Zhai, X., Kolesnikov, A., Houlsby, N., Beyer, L.: Scaling vision transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12104–12113 (2022). https://doi.org/10.1109/CVPR52688.2022.01179

62. Zhao, Y., Man, K.L., Smith, J., Siddique, K., Guan, S.U.: Improved two-stream model for human action recognition. EURASIP Journal on Image and Video Processing **2020**(1), 24 (2020). https://doi.org/10.1186/s13640-020-00501-x