GeoMFlow: Geometry-Guided Multi-Frame Optical Flow Estimation with Visual Geometry Priors

Anonymous authors
Paper under double-blind review

Abstract

Modern optical-flow estimators primarily rely on appearance correspondence and task-specific motion representations. As pixel-level motion can be viewed as the image-plane projection of underlying 3D scene and camera motion, geometry foundation models may provide scene-level structural cues that complement these representations. Motivated by this observation, we propose GeoMFlow, a lightweight geometry-conditioning framework that preserves the native matching and cost-volume pathways while injecting spatially aligned geometry features into recurrent flow refinement. Experiments on three standard optical-flow benchmarks demonstrate the effectiveness of visual geometry priors for multi-frame flow estimation. GeoMFlow further achieves an Fl-all error of **2.84%** and **ranks #1** among all optical-flow methods on the KITTI online benchmark.

1 Introduction

Optical flow estimates dense pixel-level motion between video frames and supports tasks such as motion segmentation He et al. (2026), dynamic scene reconstruction Zhu et al. (2024); Cong et al. (2026), frame interpolation Sim et al. (2021); Huang et al. (2022b), and video generation Nam et al. (2025); Liang et al. (2024). Modern estimators combine correspondence modeling with recurrent refinement, while multi-frame methods also aggregate information from neighboring frames Teed & Deng (2020); Jiang et al. (2021); Huang et al. (2022a); Shi et al. (2023b;a); Dong & Fu (2024); Sun et al. (2024); Bargatin et al. (2025). Despite these advances, appearance matching and task-specific motion representations remain central sources of evidence for estimating flow.

From a geometric perspective, optical flow is the image-plane manifestation of scene and camera motion (Fig. 1). Scene-level cues such as depth and 3D-consistent structure may therefore complement appearance correspondence. Geometry foundation models provide a source of pretrained representations that may encode such cues.

Recent advances in visual pretraining have shown that external representations can provide useful priors for optical-flow estimation Zhou et al. (2024); Poggi & Tosi (2025); Wang & Deng (2025). Geometry-oriented models extend this direction by offering representations informed by scene structure. Motivated by this, we explore how such geometry representations can be incorporated into a standard flow estimator.

GeoMFlow implements this design by preserving the native appearance-matching pathway: the matching encoder and cost volumes provide correspondence evidence, while a frozen geometry encoder extracts a spatial representation from the reference frame. The geometry features are aligned to the flow context grid and introduced as a complementary structural prior during recurrent inference; neighboring frames remain inputs to the optical-flow branch. FiLM-style modulation Perez et al. (2018) provides a lightweight means of conditioning context features while keeping the pretrained correspondence representation intact. Since the conditioning is applied only to the context branch, it can be implemented as a lightweight plug-in without modifying the backbone-specific matching mechanism, making the interface naturally compatible with modern optical-flow architectures that separate correspondence estimation from iterative refinement.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

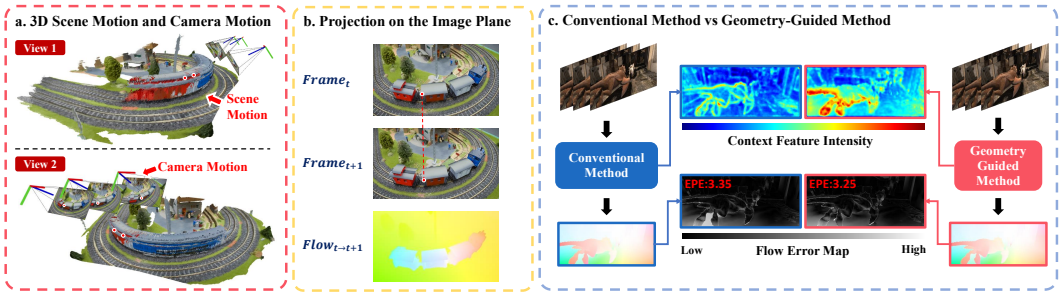


Figure 1: Optical flow from a geometric perspective. (a,b) Pixel motion reflects 3D scene and camera motion projected onto the image plane. (c) GeoMFlow preserves appearance-based correspondence as evidence and introduces scene-level geometry as a complementary prior for flow refinement.

GeoMFlow is evaluated on three standard optical flow benchmarks: MPI-Sintel Butler et al. (2012), KITTI-2015 Menze & Geiger (2015), and Spring Mehl et al. (2023). Experimental results show that pretrained visual geometry representations provide strong complementary cues for multi-frame optical flow estimation, yielding consistent gains in accuracy and cross-benchmark generalization. Remarkably, GeoMFlow achieves **rank #1** among all published optical flow methods on the KITTI online benchmark. In summary, our main contributions are:

- We conduct a controlled comparison of geometry-oriented, generic, and depth-oriented pretrained representations under a shared optical-flow framework.
- We introduce a lightweight, spatially aligned geometry-conditioning interface for modern optical-flow estimators, which preserves the native correspondence pathway and incorporates geometry as a complementary prior for recurrent refinement without modifying backbone-specific matching.
- We demonstrate the effectiveness of GeoMFlow on Sintel, KITTI, and Spring, with consistent accuracy improvements and strong cross-benchmark generalization. GeoMFlow further achieves **rank #1** among published optical-flow methods on the KITTI online benchmark.

2 Related Works

2.1 Optical Flow Estimation

Recent advancements in optical flow estimation have been driven by recurrent refinement, improved correspondence modeling, and stronger representation learning. RAFT Teed & Deng (2020) established the recurrent all-pairs correlation paradigm, becoming a strong baseline for modern optical flow methods. GMA Jiang et al. (2021) extended RAFT by introducing global motion aggregation to improve motion propagation in ambiguous regions. MatchFlow Dong et al. (2023) revisited optical flow from a geometric matching perspective, improving correspondence learning through matching-oriented pretraining. FlowFormer++ Shi et al. (2023b) strengthened cost-volume representations with masked cost-volume autoencoding. More recently, SAMFlow Zhou et al. (2024) incorporated segmentation-aware priors through SAM Kirillov et al. (2023) features, while FlowDiffuser Luo et al. (2024) explored diffusion-based flow generation instead of direct regression.

2.2 Multi-Frame Optical Flow Estimation

Multi-frame optical flow estimation uses temporal information from video sequences. VideoFlow Shi et al. (2023a) extends flow estimation with tri-frame prediction and motion propagation. MemFlow Dong & Fu (2024) reuses historical motion information, while StreamFlow Sun et al. (2024) streamlines feature and correlation computations across video frames. MEMFOF Bargatin et al. (2025) develops a memory-efficient RAFT-style multi-frame framework. These approaches primarily use appearance-based matching, temporal aggregation, or segmentation-aware features. We examine geometry-oriented representations as an additional source of scene-level structure.

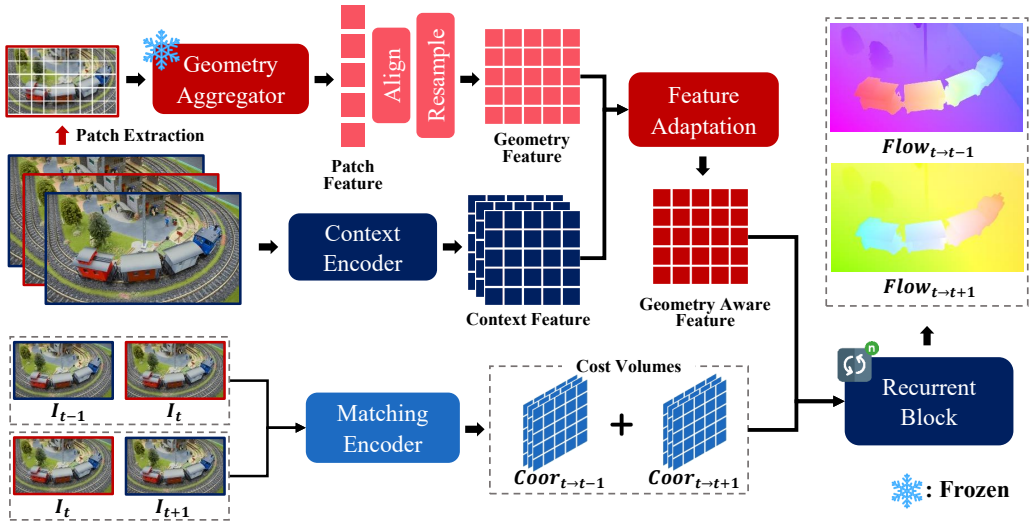


Figure 2: Overview of GeoMFlow. The native matching and bidirectional cost-volume pathways are preserved, while a frozen geometry encoder extracts a spatial prior from the center frame for context conditioning. Recurrent refinement combines correspondence evidence with geometry-aware context to estimate backward and forward flow.

2.3 Geometry and Pretrained Visual Priors

Optical-flow systems increasingly use pretrained representations as sources of additional visual information. Generic visual features, segmentation-aware cues such as those used by SAMFlow Zhou et al. (2024), matching-oriented pretraining such as MatchFlow Dong et al. (2023), and geometry-oriented representations capture different biases. WAFT Wang & Deng (2025) studies pretrained input encoders, including depth-pretrained Depth Anything V2 (DAv2) Yang et al. (2024) and DINOv3 Siméoni et al. (2025), within an iterative warping-based estimator that replaces cost-volume lookup with feature warping. FlowSeek Poggi & Tosi (2025) incorporates single-image depth foundation models into optical flow: depth-model features enrich the image features used for matching, predicted depth can condition the context, and inverse depth defines low-dimensional motion bases. GeoMFlow also studies depth- and geometry-oriented pretrained representations, introducing them through spatial context conditioning while retaining its original matching features and cost-volume pathway. Together, these works explore distinct ways of using pretrained visual and geometric information in flow estimation.

In 3D vision, DUSt3R Wang et al. (2024a) predicts point maps and related geometric quantities, while VGGT Wang et al. (2025) provides a pretrained representation learned for multi-view geometry tasks. We examine geometry-oriented representations as a source of scene structure for flow, using context conditioning alongside the established correspondence pathway.

3 Methods

3.1 Overview

As illustrated in Fig. 2, GeoMFlow follows a recurrent optical-flow estimator with a context encoder, a matching-feature encoder, and a recurrent refinement block. Given three consecutive frames I_{t-1}, I_t, I_{t+1}, the context encoder produces hidden states and context features, while the matching encoder constructs cost volumes between the center frame and its neighbors. These native matching features and cost volumes are preserved. A frozen geometry encoder processes the center (reference) frame only; the neighboring frames remain available to the optical-flow branch. Its geometry representation is aligned to the flow grid and used to condition context features. Thus, correspondence provides evidence and geometry provides a complementary structural prior for the existing bidirectional three-frame formulation.

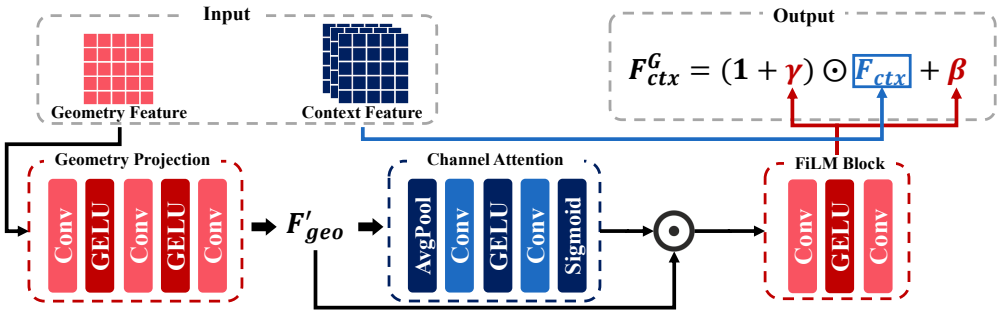


Figure 3: Geometry conditioning for context features. The projected geometry map generates spatially varying FiLM parameters, with the scale term bounded before modulating the context feature.

3.2 Geometry Feature Encoder

We extract features from the center frame I_t using a frozen pretrained geometry aggregator Wang et al. (2025); the neighboring frames remain inputs to the optical-flow branch. After resizing, I_t is divided into non-overlapping 14×14 patches and processed by the aggregator. The resulting tokens are reshaped into a spatial feature map:

$$\mathbf{Z}_g = \mathcal{G}(I_t), \qquad \mathbf{G}_p = \mathrm{reshape}(\mathbf{Z}_g) \in \mathbb{R}^{H_p \times W_p \times C_g},$$

where $\mathcal{G}$ is frozen and H_p, W_p, and C_g denote the patch-grid dimensions and feature channels.

To align this map with the optical-flow context grid at $1/16$ resolution, we bilinearly resample it at the flow-cell centers:

$$\mathbf{F}_{geo} = \mathrm{Resample}(\mathbf{G}_p, \Omega_{flow}) \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times C_g},$$

where H and W are the optical-flow input dimensions, and Ω_{flow} maps flow-grid locations to the geometry feature map. The aligned feature is then passed to the feature-adaptation module.

3.3 Feature Adaptation

Given three consecutive frames $\{I_{t-1}, I_t, I_{t+1}\}$, we first concatenate them along the channel dimension and feed them into the context encoder. The context encoder produces a feature representation that is split into an initial hidden state and a context feature:

$$[\mathbf{h}_0, \mathbf{F}_{ctx}] = \mathcal{E}_c([I_{t-1}, I_t, I_{t+1}]),$$

where $\mathcal{E}_c(\cdot)$ denotes the context encoder, $\mathbf{h}_0$ is the initial hidden state for recurrent refinement, and $\mathbf{F}_{ctx} \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times C}$ is the optical-flow context feature.

The adaptation module introduces geometry conditioning while preserving the pretrained flow representation. Specifically, the aligned geometry feature $\mathbf{F}_{geo}$ is injected into the context feature $\mathbf{F}_{ctx}$ (Fig. 3). We first project the geometry feature and apply channel-wise recalibration:

$$\mathbf{F}'_{geo} = \phi_g(\mathbf{F}_{geo}), \quad \tilde{\mathbf{F}}_{geo} = \mathbf{F}'_{geo} \odot \sigma\left(\phi_a(\mathrm{GAP}(\mathbf{F}'_{geo}))\right),$$

where $\phi_g(\cdot)$ denotes the geometry projection module, $\phi_a(\cdot)$ denotes the channel-attention module, $\mathrm{GAP}(\cdot)$ is global average pooling, $\sigma(\cdot)$ is the sigmoid function, and $\odot$ denotes element-wise multiplication. GAP is used only to compute channel-recalibration weights, which are applied back to the full spatial geometry map; thus, the spatial structure of the geometry representation is retained.

We then use FiLM-style modulation Perez et al. (2018) to condition the optical-flow context feature. The recalibrated geometry feature is transformed into a raw scale term $\hat{\gamma}$ and a shift term β:

$$[\hat{\gamma}, \beta] = \phi_f(\tilde{\mathbf{F}}_{geo}),$$

where $\phi_f(\cdot)$ is implemented with lightweight convolutional layers operating on the spatial geometry map. Both $\hat{\gamma}(x, y)$ and $\beta(x, y)$ vary spatially, keeping the conditioning aligned with the flow context. To stabilize training, we bound the scale modulation as

$$\gamma = 0.1 \cdot \tanh(\hat{\gamma}).$$

The final geometry-conditioned context feature is then computed as

$$\mathbf{F}_{ctx}^{G} = (1+\gamma) \odot \mathbf{F}_{ctx} + \beta.$$

This adaptation introduces spatially aligned structural guidance while retaining the original correspondence evidence. The conditioned context feature $\mathbf{F}_{ctx}^{G}$ is used in the subsequent recurrent refinement.

3.4 Bi-directional Recurrent Flow Refinement

Following the bidirectional multi-frame formulation Shi et al. (2023a); Bargatin et al. (2025), the model estimates backward flow $\mathbf{Flow}_{t\to t-1}$ and forward flow $\mathbf{Flow}_{t\to t+1}$ from three consecutive frames $\{I_{t-1}, I_t, I_{t+1}\}$. Shared matching features $\{\mathbf{F}_{t-1}^{m}, \mathbf{F}_{t}^{m}, \mathbf{F}_{t+1}^{m}\}$ from encoder $\mathcal{E}_m(\cdot)$ construct the correlation volumes:

$$\mathbf{C}_{t\to t-1} = \mathrm{Corr}(\mathbf{F}_{t}^{m}, \mathbf{F}_{t-1}^{m}), \quad \mathbf{C}_{t\to t+1} = \mathrm{Corr}(\mathbf{F}_{t}^{m}, \mathbf{F}_{t+1}^{m}).$$

The correlation volumes provide correspondence evidence for the two flow estimates, while geometry conditions the recurrent context used to refine them.

The recurrent refinement starts from an initial hidden state and initial bidirectional flow estimates. At each iteration, the current flows are used to index the corresponding correlation volumes and retrieve local matching costs. The retrieved correlation features, current flow estimates, hidden state, and geometry-aware context feature $\mathbf{F}_{ctx}^{G}$ are fed into the recurrent update block:

$$\mathbf{h}_{k+1} = \mathcal{U}\left(\mathbf{h}_k, \mathbf{F}_{ctx}^{G}, \mathbf{c}_{t\to t-1}^{k}, \mathbf{c}_{t\to t+1}^{k}, \mathbf{Flow}_{t\to t-1}^{k}, \mathbf{Flow}_{t\to t+1}^{k}\right).$$

The updated hidden state is decoded into residual flow increments $\Delta\mathbf{Flow}^{k}$, which are added to the current flow estimates:

$$\mathbf{Flow}_{d}^{k+1} = \mathbf{Flow}_{d}^{k} + \Delta\mathbf{Flow}_{d}^{k}, \quad d \in \{t\to t-1, t\to t+1\}.$$

The geometry-aware context feature remains fixed during the recurrent iterations and provides structural guidance for hidden-state updating. After K iterations, the final low-resolution bidirectional flows are upsampled to the original image resolution using a learned convex upsampling module.

3.5 Optimization

Following prior recurrent optical-flow methods Bargatin et al. (2025); Teed & Deng (2020), we supervise all intermediate flow predictions produced during recurrent refinement. Given the ground-truth bidirectional flows $\mathbf{F}_{t\to t-1}^{gt}$ and $\mathbf{F}_{t\to t+1}^{gt}$, the model predicts a sequence of flow estimates $\{\mathbf{F}_{t\to t-1}^{k}, \mathbf{F}_{t\to t+1}^{k}\}_{k=1}^{K}$ over K refinement iterations. We apply a weighted sequence loss:

$$\mathcal{L}_{flow} = \sum_{k=1}^{K} \lambda^{K-k} \sum_{d\in\{t\to t-1,\, t\to t+1\}} \left\|\mathbf{F}_{d}^{k} - \mathbf{F}_{d}^{gt}\right\|_1,$$

where λ is the exponential weighting factor used to assign relatively larger weights to later refinement iterations.

4 Experiments

4.1 Experimental Setup

Evaluation Datasets and Metrics. We evaluate GeoMFlow on three standard benchmarks: MPI-Sintel Butler et al. (2012), a synthetic dataset with complex motion; KITTI-2015 Menze & Geiger (2015), a real-world autonomous driving dataset; and Spring Mehl et al. (2023), a high-resolution optical flow benchmark. We report the endpoint error (EPE), 1-pixel error rate (1px), Fl (KITTI-15), and WAUC as comparison metrics.

Network and Training. GeoMFlow is implemented in PyTorch Paszke et al. (2017), and all experiments are conducted on NVIDIA RTX 4090 GPUs unless otherwise noted. The geometry

Table 1: Benchmark results on MPI-Sintel and KITTI-15, with * indicating multi-frame methods. Best and Second results are highlighted.

Methods		Sintel Clean			Sintel Final			KITTI-15	
		Mat.	Unm.	All	Mat.	Unm.	All	Non-Occ	Fl-all
RAFT	[ECCV2020]	0.623	9.647	1.609	1.405	14.680	2.855	3.07	5.10
GMFlow	[CVPR2022]	0.653	10.555	1.736	1.319	15.797	2.902	3.80	9.32
GMA	[ICCV2021]	0.582	7.963	1.388	1.241	12.501	2.470	-	5.15
SKFlow	[NIPS2022]	0.567	7.251	1.298	1.138	11.415	2.261	-	4.84
MatchFlow	[CVPR2023]	0.431	7.130	1.164	1.061	13.070	2.373	2.77	4.63
FlowFormer++	[CVPR2023]	0.390	6.635	1.073	0.878	10.627	1.943	-	4.52
SAMFlow	[AAAI2024]	0.384	5.966	0.995	1.036	10.600	2.080	-	4.49
FlowDiffuser	[CVPR2024]	0.376	6.231	1.016	0.966	10.666	2.026	2.82	4.17
DPFlow	[CVPR2025]	0.394	6.361	1.046	0.906	10.685	1.975	2.12	3.56
WAFT	[ICLR2026]	0.347	5.830	0.946	1.006	10.315	2.023	2.13	3.56
MFCFlow*	[WACV2023]	0.653	8.335	1.493	1.326	12.805	2.579	-	5.00
TransFlow*	[CVPR2023]	0.357	6.770	1.058	0.986	10.960	2.076	-	4.32
StreamFlow*	[NIPS2024]	0.381	6.416	1.041	0.824	10.435	1.874	2.45	4.24
VideoFlow*	[ICCV2023]	0.389	6.023	1.005	0.812	9.054	1.713	-	4.44
MemFlow*	[CVPR2024]	0.426	6.091	1.046	0.931	9.928	1.914	2.56	4.10
MemFlow-T*	[CVPR2024]	0.430	6.384	1.081	0.874	9.710	1.840	2.45	3.88
MEMFOF*	[ICCV2025]	0.397	5.587	0.963	0.877	10.302	1.907	1.97	2.94
GeoMFlow*	(Ours)	0.383	5.518	0.946	0.809	9.756	1.792	1.90	2.84

aggregator is initialized with pretrained weights from Wang et al. (2025) and kept frozen during training. Following Wang et al. (2024b); Bargatin et al. (2025), we first pretrain the model on TartanAir Wang et al. (2020) and FlyingThings Mayer et al. (2016), referred to as the T stage, and perform zero-shot evaluation on the training splits of Sintel Butler et al. (2012) and KITTI-2015 Menze & Geiger (2015). We then conduct the TSKH fine-tuning stage with a mixture of FlyingThings, Sintel, KITTI-2015, and HD1K Kondermann et al. (2016). Prior to submission to the online benchmarks, we apply lightweight dataset-specific fine-tuning on each target dataset. Additionally, we adopt an incremental training strategy Wang et al. (2022); Pfeiffer et al. (2021). For each stage, the context encoder and the matching feature encoder are initialized with weights from MEMFOF Bargatin et al. (2025) and kept frozen during training, while only the feature adaptation module and the recurrent refinement components are optimized. This efficient strategy reduces training cost and preserves the pretrained appearance and geometry representations, while enabling the flow estimator to adapt to geometry-guided context features. More details can be found in the appendix.

4.2 Main Benchmark Results

Results on Sintel and KITTI. Tab. 1 compares GeoMFlow with two-frame and multi-frame flow estimators on Sintel and KITTI. GeoMFlow obtains an overall EPE of $\mathbf{0.946}$ on Sintel Clean and reports $\mathbf{1.90}$ Non-Occ and $\mathbf{2.84}$ Fl-all on KITTI. Relative to MEMFOF Bargatin et al. (2025), the reported EPE changes are $\mathbf{1.7}\%$ and $\mathbf{6.0}\%$ on Sintel Clean and Final, and the Fl-all reduction is $\mathbf{3.4}\%$. At the reported KITTI online benchmark snapshot, GeoMFlow ranked #1 among the compared methods. These results characterize the full estimator; the feature-source comparison below examines the role of geometry representations.

Results on Spring. Tab. 2 reports the evaluation results on the Spring benchmark. GeoMFlow achieves competitive Spring performance, improving EPE and WAUC over MEMFOF while matching its Fl. We also report the inference cost on 1080p-resolution inputs.

Table 2: Benchmark results on Spring, with * indicating multi-frame methods. Runtime and maximum GPU memory usage were evaluated using an NVIDIA RTX 3090 GPU.

Methods	Inference Cost(1080p)		Spring(test)			
	Memory,GB	Runtime,ms	EPE↓	1px↓	WAUC↑	Fl↓
CrocoFlow Weinzaepfel et al. (2023)	2.01	6524	0.498	4.565	93.660	1.508
SEA-RAFT (S) Wang et al. (2024b)	8.15	205	0.377	3.904	94.182	1.389
SEA-RAFT (M) Wang et al. (2024b)	8.19	286	0.363	3.686	94.534	1.347
MemFlow* Dong & Fu (2024)	8.08	885	0.471	4.482	93.855	1.416
StreamFlow* Sun et al. (2024)	18.97	929	0.467	4.152	94.404	1.424
MEMFOF* Bargatin et al. (2025)	2.09	472	0.355	3.289	95.186	1.238
GeoMFlow*	5.95	489	0.351	3.439	95.208	1.238

Table 3: Zero-shot evaluation on the Sintel and KITTI training sets, with * indicating multi-frame methods.

Input Data	Methods		Sintel(train)		KITTI-15(train)	
			Clean	Final	EPE	Fl-all
Pairwise	RAFT Teed & Deng (2020)	[ECCV2020]	1.41	2.70	5.04	17.4
	MatchFlow Dong et al. (2023)	[CVPR2023]	1.03	2.41	4.08	15.6
	FlowFormer++ Shi et al. (2023b)	[CVPR2023]	0.90	2.31	3.93	14.1
	SAMFlow Zhou et al. (2024)	[AAAI2024]	0.87	2.11	3.47	12.3
	FlowDiffuser Luo et al. (2024)	[CVPR2024]	0.86	2.19	3.61	11.8
	DPFlow Morimitsu et al. (2025)	[CVPR2025]	1.02	2.26	3.35	11.1
	WAFT Wang & Deng (2025)	[ICLR2026]	1.01	2.49	3.28	10.9
	FlowSeek Poggi & Tosi (2025)	[ICCV2025]	1.04	2.18	3.31	11.2
Multi-frame	StreamFlow* Sun et al. (2024)	[NIPS2024]	0.87	2.11	3.85	12.6
	VideoFlow* Shi et al. (2023a)	[ICCV2023]	1.03	2.19	3.96	15.3
	MemFlow-T* Dong & Fu (2024)	[CVPR2024]	0.86	2.06	3.38	12.8
	MEMFOF* Bargatin et al. (2025)	[ICCV2025]	1.20	3.91	2.93	9.93
	GeoMFlow*	(Ours)	1.03	2.81	2.47	8.82

4.2.1 Zero-Shot Evaluation

We perform zero-shot evaluation on the Sintel and KITTI training sets using the T-stage model (Tab. 3). GeoMFlow reports EPE $\mathbf{2.47}$ and Fl-all $\mathbf{8.82}$ on KITTI. Relative to MEMFOF Bargatin et al. (2025), the reported changes are $\mathbf{15.7}\%$ and $\mathbf{11.2}\%$ on KITTI EPE and Fl-all, and $\mathbf{14.2}\%$ and $\mathbf{28.1}\%$ on Sintel Clean and Final.

4.3 Generic, Depth-Oriented, and Geometry-Oriented Pretrained Features

We compare DINOv2 Oquab et al. (2023), the Depth Anything V2 (DAv2) Yang et al. (2024), and VGGT features using the same MEMFOF-based flow backbone, feature-adaptation interface, and training protocol. Sintel and KITTI results are evaluated on their training splits without dataset-specific fine-tuning; Spring results are reported on its training split. This setup compares a generic visual representation with depth- and geometry-oriented representations while holding the flow estimator and integration method fixed. Tab. 4 reveals a clear difference among pretrained representations. DINOv2 and DAv2 improve several metrics, while VGGT provides the most consistent gains across all three benchmarks. These comparisons show that the choice of pretrained representation matters. Generic and depth-oriented features provide mixed but measurable gains, while VGGT yields the most consistent improvements under the shared integration setup.

Table 4: Controlled comparison of pretrained feature sources using the same MEMFOF-based optical-flow backbone and geometry-conditioning interface. Sintel and KITTI results are zero-shot evaluations on their training splits; Spring results are on its training split. Lower is better for all metrics except WAUC.

Settings	Sintel		KITTI		Spring			
	clean	final	EPE	Fl-all	1px	EPE	Fl	WAUC
Baseline(w/o Geo.)	1.20	3.91	2.93	9.93	3.55	0.381	1.040	92.32
GeoMFlow-DINOv2	1.39	3.06	2.98	9.35	3.34	0.348	0.947	92.67
GeoMFlow-DAv2	1.23	2.91	2.97	8.90	3.29	0.353	0.928	92.70
GeoMFlow-VGGT	1.03	2.81	2.47	8.82	3.19	0.329	0.899	92.88

Table 5: Ablation Study on Spring training set.

Settings		EPE↓				1px↓				Other	
		avg	s0-10	s10-40	s40+	avg	s0-10	s10-40	s40+	WAUC↑	Fl↓
Baseline(w/o Geo.)		0.381	0.154	0.959	6.866	3.553	1.995	12.218	34.413	92.321	1.040
Train Stgy.	Full FT	0.377	0.155	0.958	6.678	3.415	1.878	11.978	33.819	92.237	1.029
	Frz M-Enc.	0.360	0.146	0.894	6.516	3.332	1.806	11.745	33.772	92.598	0.984
	Frz C-Enc.	0.362	0.144	0.893	6.670	3.368	1.841	11.841	33.674	92.478	0.990
	Frz Both	0.329	0.124	0.794	6.366	3.197	1.683	11.530	33.458	92.879	0.899
Fusion Type	Mat. Feat.	0.373	0.145	0.913	6.996	3.573	1.959	12.594	35.433	92.030	1.026
	Add	0.370	0.156	0.899	6.521	3.441	1.923	11.823	33.678	91.907	1.027
	FiLM	0.329	0.124	0.794	6.366	3.197	1.683	11.530	33.458	92.879	0.899

4.4 Ablation: Where Geometry Enters the Flow Estimator

We analyze the training strategy and where geometry enters the estimator using the Spring training set and the model from stage TSKH (Tab. 5). The baseline has no geometry adaptation.freezing both pretrained flow encoders (matching and context) gives the best overall result among the tested training settings. In the fusion comparison, modifying matching features before cost-volume construction performs worse than context-level conditioning. Additive context fusion also underperforms FiLM-style modulation in this experiment. Together, these results favor introducing geometry through the recurrent context while retaining the original correspondence features.

4.5 Qualitative Comparison

We present qualitative comparisons with MEMFOF on the Sintel and KITTI datasets. Fig. 4 shows representative results on the Sintel Final split. As highlighted by the red circles, our method better captures the flow of fine-grained structures(the hair region), which is also reflected in the error maps, where brighter regions denote larger errors. On KITTI, as shown in Fig. 5, MEMFOF incorrectly blends the motion of the white wall and the vehicle body, whereas our method, aided by geometric priors, separates and estimates their motion fields more accurately. These results suggest that geometric priors provide complementary cues for optical flow estimation, especially when 2D appearance information is ambiguous, such as in geometrically complex or textureless regions.



Figure 4: Qualitative flow predictions and error maps on the Sintel test set.



Figure 5: Qualitative flow predictions and error maps on the KITTI-2015 test set.

Table 6: Zero-shot geometry transfer on the Sintel and KITTI training splits

Backbone	Sintel		KITTI	
	Clean	Final	EPE	Fl
FlowFormer++	0.90	2.31	3.93	14.13
Geom Enhanced.	0.81 ↓**10.0%**	2.05 ↓**11.2%**	3.24 ↓**17.6%**	12.40 ↓**12.2%**
MEMFOF	1.20	3.91	2.93	9.93
Geom Enhanced.	1.03 ↓**14.1%**	2.81 ↓**28.1%**	2.47 ↓**15.6%**	8.82 ↓**11.2%**

4.6 Architecture Generality of Geometry Conditioning

To assess whether context-branch conditioning extends beyond MEMFOF, we apply the same interface to FlowFormer++ Shi et al. (2023b), a two-frame estimator, and compare both models with and without geometry conditioning. In each case, geometry conditions the context used for iterative refinement while the backbone's matching representation remains unchanged. Table 6 reports zero-shot results on the Sintel and KITTI training splits; percentages indicate relative error reductions over the corresponding backbone. Geometry conditioning reduces every reported error, with gains ranging from 10.0% to 17.6% for FlowFormer++ and from 11.2% to 28.1% for MEMFOF. The consistent gains across FlowFormer++ and MEMFOF further indicate that this lightweight plug-in design can be transferred between the tested two-frame and multi-frame architectures while preserving their native matching representations.

5 Limitations

GeoMFlow introduces geometry through recurrent context conditioning while preserving the original correspondence pathway. This design avoids perturbing the pretrained matching representation, while leaving the direct integration of geometric cues into correspondence estimation as a direction for future exploration. GeoMFlow also relies on a pretrained geometry representation and does not explicitly model the reliability of the resulting prior. Future work may therefore explore both geometry-aware matching and reliability-aware conditioning.

6 Conclusion

We investigated geometry-oriented pretrained representations for optical flow and proposed GeoMFlow, which introduces spatially aligned geometry cues into recurrent refinement while preserving the native correspondence pathway. Controlled comparisons with generic and depth-oriented pretrained features show that geometry-oriented representations provide effective complementary cues for flow estimation. The same conditioning interface also improves both MEMFOF and FlowFormer++, indicating its applicability across the tested multi-frame and two-frame architectures. GeoMFlow achieves strong performance on MPI-Sintel, KITTI-2015, and Spring, including an Fl-all error of **2.84%** and **rank #1** among published optical-flow methods on the KITTI online benchmark. Future work will explore geometry-aware matching and reliability-aware geometry conditioning.

AI use statement

ChatGPT (OpenAI) was used to assist with editing and restructuring manuscript prose, drafting related-work and figure/table descriptions, and summarizing experimental results supplied by the

authors. It was not used to generate or modify experimental data, implement or run the models, or produce the reported measurements. The authors reviewed the AI-assisted text and take responsibility for the final manuscript and its claims.

Ethics statement

This study uses established optical-flow benchmarks and pretrained representations cited in the paper. It does not recruit human participants or collect new personal data. Although some benchmark videos depict people, the method estimates pixel-level motion and does not identify individuals or infer sensitive attributes. The datasets and pretrained models are used subject to their respective terms of access and use. We are not aware of additional ethics concerns specific to this work.

Reproducibility statement

Section 3 describes the model and training objective, and Section 4 reports the datasets, evaluation protocols, metrics, and experimental results. Appendix A details the training stages, data mixture, preprocessing, crop sizes, optimization steps, recurrent iterations, and hardware. The current submission package does not include source code or model checkpoints.

References

Vladislav Bargatin, Egor Chistov, Alexander Yakovenko, and Dmitriy Vatolin. Memfof: High-resolution training for memory-efficient multi-frame optical flow estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8187–8196, 2025.

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

Zhongxiao Cong, Qitao Zhao, Minsik Jeon, and Shubham Tulsiani. Flow3r: Factored flow prediction for scalable visual geometry learning. In *CVPR*, 2026.

Qiaole Dong and Yanwei Fu. Memflow: Optical flow estimation and prediction with memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19068–19078, 2024.

Qiaole Dong, Chenjie Cao, and Yanwei Fu. Rethinking optical flow from geometric matching consistent perspective. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 1337–1347, 2023.

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

Xiankang He, Peile Lin, Ying Cui, Dongyan Guo, Chunhua Shen, and Xiaoqin Zhang. GeoMotion: Rethinking motion segmentation via latent 4d geometry. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'26)*, 2026.

Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer: A transformer architecture for optical flow. In *European conference on computer vision*, pp. 668–685. Springer, 2022a.

Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *European conference on computer vision*, pp. 624–642. Springer, 2022b.

Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9772–9781, 2021.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023.

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

Feng Liang, Bichen Wu, Jialiang Wang, Licheng Yu, Kunpeng Li, Yinan Zhao, Ishan Misra, Jia-Bin Huang, Peizhao Zhang, Peter Vajda, et al. Flowvid: Taming imperfect optical flows for consistent video-to-video synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8207–8216, 2024.

Ao Luo, Xin Li, Fan Yang, Jiangyu Liu, Haoqiang Fan, and Shuaicheng Liu. 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.

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

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

Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3061–3070, 2015.

Henrique Morimitsu, Xiaobin Zhu, Roberto M Cesar, Xiangyang Ji, and Xu-Cheng Yin. Dpflow: Adaptive optical flow estimation with a dual-pyramid framework. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 17810–17820, 2025.

Hyelin Nam, Jaemin Kim, Dohun Lee, and Jong Chul Ye. Optical-flow guided prompt optimization for coherent video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 7837–7846, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 724–732, 2016.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th conference of the European chapter of the association for computational linguistics: main volume*, pp. 487–503, 2021.

Matteo Poggi and Fabio Tosi. Flowseek: Optical flow made easier with depth foundation models and motion bases. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5667–5679. IEEE, 2025.

Xiaoyu Shi, Zhaoyang Huang, Weikang Bian, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Videoflow: Exploiting temporal cues for multi-frame optical flow estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12469–12480, 2023a.

Xiaoyu Shi, Zhaoyang Huang, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer++: Masked cost volume autoencoding for pretraining optical flow estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1599–1610, 2023b.

Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. Xvfi: extreme video frame interpolation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 14489–14498, 2021.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Shangkun Sun, Jiaming Liu, Huaxia Li, Guoqing Liu, Thomas H Li, and Wei Gao. Streamflow: streamlined multi-frame optical flow estimation for video sequences. *Advances in neural information processing systems*, 37:9205–9228, 2024.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pp. 402–419. Springer, 2020.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20697–20709, 2024a.

Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. 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. IEEE, 2020.

Yihan Wang and Jia Deng. Waft: Warping-alone field transforms for optical flow. *arXiv e-prints*, pp. arXiv–2506, 2025.

Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. In *European Conference on Computer Vision*, pp. 36–54. Springer, 2024b.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 139–149, 2022.

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

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in neural information processing systems*, 37:21875–21911, 2024.

Shili Zhou, Ruian He, Weimin Tan, and Bo Yan. Samflow: Eliminating any fragmentation in optical flow with segment anything model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 7695–7703, 2024.

Ruijie Zhu, Yanzhe Liang, Hanzhi Chang, Jiacheng Deng, Jiahao Lu, Wenfei Yang, Tianzhu Zhang, and Yongdong Zhang. Motiongs: Exploring explicit motion guidance for deformable 3d gaussian splatting. *Advances in Neural Information Processing Systems*, 37:101790–101817, 2024.

A Implementation Details

This section provides training and inference details for the geometry-conditioned flow estimator described in the main text. We follow the training protocols of Bargatin et al. (2025); Wang et al. (2024b), omitting FlyingChairs Dosovitskiy et al. (2015) because it contains only two frames. For all datasets except Spring Mehl et al. (2023), images and flows are upsampled by a factor of two to support the crop sizes listed in Table 7.

The initial TartanAir stage trains the flow estimator without geometry conditioning. In the subsequent geometry-conditioned stages, the geometry aggregator and the pretrained context and matching encoders remain frozen; only the feature-adaptation and recurrent-refinement modules are optimized. During the TSKH stage, the training mixture contains S (0.32), T (0.31), K (0.12), and H (0.24), following Bargatin et al. (2025); Wang et al. (2024b). For efficiency, images are downsampled by a factor of two before geometry feature extraction. Inputs are padded when a crop exceeds the frame dimensions or is not divisible by both patch sizes (16 for the flow grid and 14 for the geometry encoder), allowing the two feature maps to be aligned.

At inference, we use $K = 8$ recurrent refinement iterations for all datasets except Sintel, where $K = 16$. Training and fine-tuning use NVIDIA RTX 4090 GPUs (24 GB), except for Spring fine-tuning, which uses an RTX 5090 GPU (32 GB). Table 7 summarizes the stages, crop sizes, iteration counts, and training steps.

Table 7: Overview of our training procedure. Dataset abbreviations are as follows: T: Things, S: Sintel, K: KITTI-2015, and H: HD1K. K denotes the number of recurrent refinement iterations.

Stage	Weights	Datasets	Scale	Crop Size	K	Steps	Freeze-FT. Steps
TartanAir	-	TartanAir	2	[480, 960]	4	225k	-
Things (T)	TartanAir	T	2	[864, 1920]	4	100k	20k
TSKH	Things	T+S+K+H	2	[864, 1920]	4	220k	60k
Sintel-ft	TSKH	S	2	[872, 1920]	8	12.5k	40k
KITTI-ft	TSKH	K	2	[750, 1920]	8	2.5k	10k
Spring-ft	TSKH	Spring	1	[1080, 1920]	8	60k	40k

B Additional Qualitative Results

Figures 6, 8, and 9 show supplementary flow predictions and error maps on Sintel, KITTI, and Spring. Brighter error-map regions indicate larger endpoint errors; red boxes and circles mark selected regions for visual comparison. The examples include texture-poor areas, fine structures, and scene boundaries, where the geometry-conditioned model and baseline produce different estimates. These visualizations complement the aggregate results in the main-text tables and offer qualitative examples of geometry as an additional context cue.

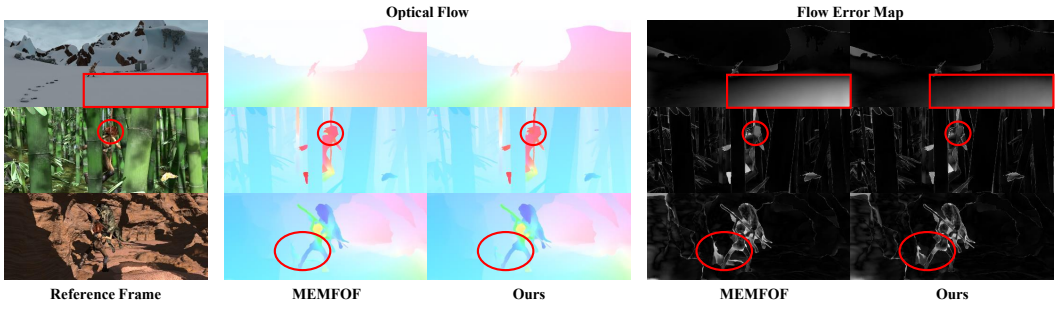


Figure 6: Qualitative flow predictions and error maps on the Sintel test set.

Evaluation ground truth: All pixels Evaluation area: All pixels

	Method	Setting	Code	Fl-bg	Fl-fg	Fl-all	Density	Runtime	Environment	Compare
1	GeoMFlow			2.57 %	**4.20 %**	**2.84 %**	100.00 %	0.4 s	GPU @ 2.5 Ghz (Python)	
2	MS-RAFT-3D+		code	2.22 %	5.99 %	2.85 %	100.00 %	3 s	GPU @ 2.5 Ghz (Python)	
	J. Schmid, A. Jahedi, N. Senn and A. Bruhn: MS-RAFT-3D: A Multi-Scale Architecture for Recurrent Image-Based Scene Flow. IEEE International Conference on Image Processing (ICIP) 2025.									
3	ARFlow			2.48 %	4.69 %	2.85 %	100.00 %	0.35 s	GPU @ 2.5 Ghz (Python)	
	J. Liu, M. Liu, S. Zhu, Y. Zhang, J. Li, M. Yang, F. Nex, H. Cheng and H. Wang: ARFlow: Auto-regressive Optical Flow Estimation for Arbitrary-Length Videos via Progressive Next-Frame Forecasting. The Fourteenth International Conference on Learning Representations 2026.									
4	VGPFlow		code	2.57 %	4.35 %	2.87 %	100.00 %	0.2 s	1 core @ 2.5 Ghz (C/C++)	
5	SplatFlow3D		code	2.27 %	6.02 %	2.89 %	100.00 %	0.1 s	1 core @ 2.5 Ghz (Python)	
	B. Wang, Y. Zhang, J. Li, Y. Yu, Z. Sun, L. Liu and D. Hu: SplatFlow: Learning Multi-frame Optical Flow via Splatting. International Journal of Computer Vision 2024.									
6	MEMFOF_t			2.49 %	4.94 %	2.90 %	100.00 %	1 s	1 core @ 2.5 Ghz (Python)	
7	MEMFOF		code	2.60 %	4.66 %	2.94 %	100.00 %	0.4 s	GPU @ 2.5 Ghz (Python)	
	V. Bargatin, E. Chistov, A. Yakovenko and D. Vatolin: MEMFOF: High-Resolution Training for Memory-Efficient Multi-Frame Optical Flow Estimation. arXiv preprint arXiv:2506.23151 2025.									
8	CamLiRAFT		code	**2.08 %**	7.37 %	2.96 %	100.00 %	1 s	GPU @ 2.5 Ghz (Python + C/C++)	
	H. Liu, T. Lu, Y. Xu, J. Liu and L. Wang: Learning Optical Flow and Scene Flow with Bidirectional Camera-LiDAR Fusion. TPAMI 2023.									
9	GFFlow_30k			2.55 %	5.13 %	2.98 %	100.00 %	0.3 s	1 core @ 2.5 Ghz (Python)	
10	FlowPainter			2.35 %	6.37 %	3.02 %	100.00 %	0.38 s	1 core @ 2.5 Ghz (Python)	
11	GFFlow_50k		code	2.57 %	5.39 %	3.04 %	100.00 %	0.3 s	1 core @ 2.5 Ghz (Python)	
12	GFFlow_ft_20k		code	2.75 %	4.68 %	3.07 %	100.00 %	0.3 s	1 core @ 2.5 Ghz (Python)	
13	CamLiFlow		code	2.31 %	7.04 %	3.10 %	100.00 %	1.2 s	GPU @ 2.5 Ghz (Python + C/C++)	
	H. Liu, T. Lu, Y. Xu, J. Liu, W. Li and L. Chen: CamLiFlow: Bidirectional Camera-LiDAR Fusion for Joint Optical Flow and Scene Flow Estimation. CVPR 2022.									
14	GeoFlow-M			2.79 %	4.81 %	3.13 %	100.00 %	1 s	1 core @ 2.5 Ghz (C/C++)	
15	FreeFlow-L		code	2.89 %	4.92 %	3.23 %	100.00 %	0.6 s	GPU @ 2.5 Ghz (Python)	
	V. Bargatin, A. Yakovenko, K. Abud and D. Vatolin: FreeFlow: A Bias-free Hierarchical Transformer for Optical Flow Estimation. European Conference on Computer Vision (ECCV) 2026.									
16	DDVM			2.90 %	5.05 %	3.26 %	100.00 %			
	S. Saxena, C. Herrmann, J. Hur, A. Kar, M. Norouzi, D. Sun and D. Fleet: The Surprising Effectiveness of Diffusion Models for Optical Flow and Monocular Depth Estimation. NeurIPS 2023.									
17	FreeFlow-M		code	3.09 %	4.58 %	3.33 %	100.00 %	0.3 s	GPU @ 2.5 Ghz (Python)	
	V. Bargatin, A. Yakovenko, K. Abud and D. Vatolin: FreeFlow: A Bias-free Hierarchical Transformer for Optical Flow Estimation. European Conference on Computer Vision (ECCV) 2026.									
18	GGR-Flow		code	2.65 %	6.89 %	3.36 %	100.00 %	0.90 s	1 core @ 2.5 Ghz (C/C++)	
19	mmFlow			2.89 %	5.97 %	3.40 %	100.00 %	1 s	1 core @ 2.5 Ghz (C/C++)	
20	CamLiRAFT-NR		code	2.76 %	6.78 %	3.43 %	100.00 %	1 s	GPU @ 2.5 Ghz (Python + C/C++)	
	H. Liu, T. Lu, Y. Xu, J. Liu and L. Wang: Learning Optical Flow and Scene Flow with Bidirectional Camera-LiDAR Fusion. arXiv preprint arXiv:2303.12017 2023.									
21	M-FUSE		code	2.66 %	7.47 %	3.46 %	100.00 %	1.3 s	GPU	
	L. Mehl, A. Jahedi, J. Schmalfuss and A. Bruhn: M-FUSE: Multi-frame Fusion for Scene Flow Estimation. Proc. Winter Conference on Applications of Computer Vision (WACV) 2023.									
22	GGR-Flow-1		code	2.75 %	7.05 %	3.47 %	100.00 %	0.90 s	1 core @ 2.5 Ghz (C/C++)	
23	RigidMask+ISF		code	2.63 %	7.85 %	3.50 %	100.00 %	3.3 s	GPU @ 2.5 Ghz (Python)	
	G. Yang and D. Ramanan: Learning to Segment Rigid Motions from Two Frames. CVPR 2021.									
24	RAFT-3D-MF			2.73 %	7.64 %	3.55 %	100.00 %	1 s	1 core @ 2.5 Ghz (C/C++)	
25	DPFlow		code	3.29 %	4.93 %	3.56 %	100.00 %	0.1 s	GPU @ 2.5 Ghz (Python)	
	H. Morimitsu, X. Zhu, R. Cesar-Jr., X. Ji and X. Yin: DPFlow: Adaptive Optical Flow Estimation with a Dual-Pyramid Framework. CVPR 2025.									

Figure 7: Archived KITTI-2015 optical-flow benchmark rankings.



Figure 8: Qualitative flow predictions and error maps on the KITTI test set.

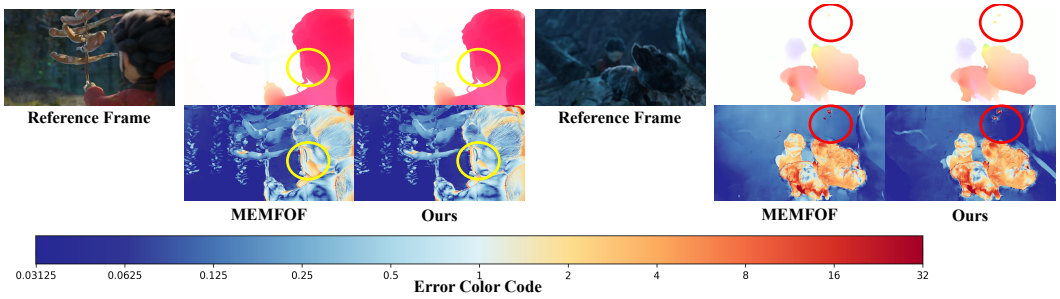


Figure 9: Qualitative flow predictions and error maps on the Spring test set.

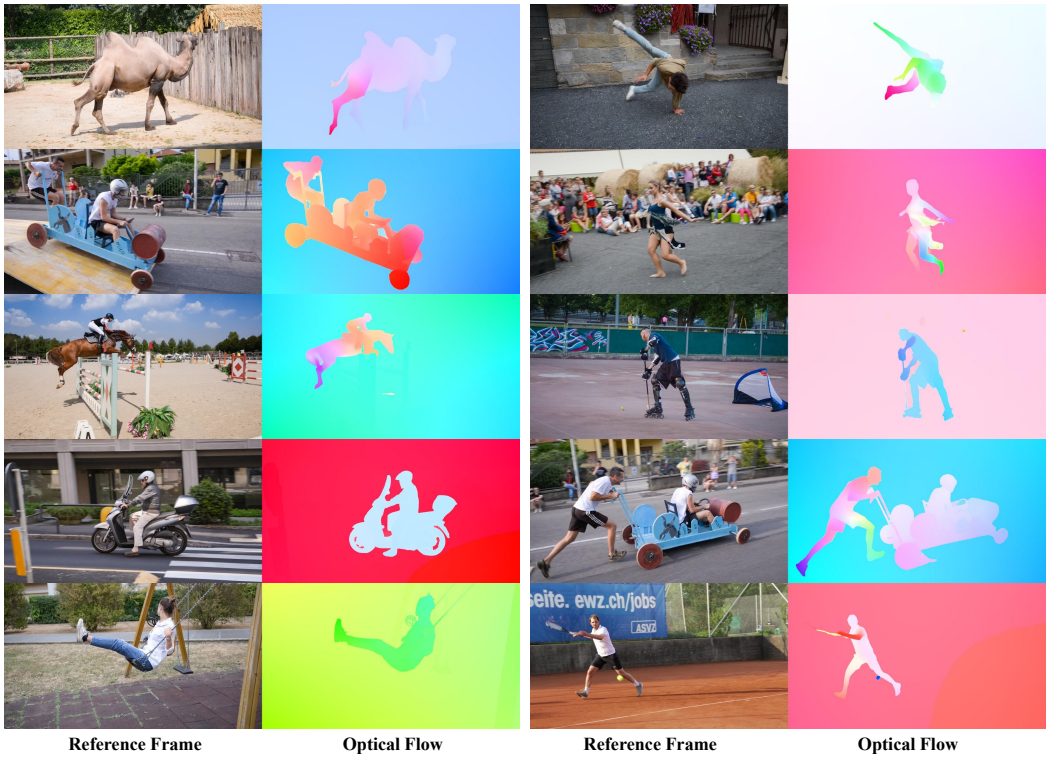


Figure 10: Zero-shot flow predictions on DAVIS Perazzi et al. (2016).

C SCREENSHOT ON KITTI BENCHMARK WEBSITE

Figure 7 reproduces the KITTI-2015 benchmark ranking snapshot used for this submission. GeoM-Flow is the highest-ranked method in this snapshot.

D ZERO-SHOT QUALITATIVE RESULTS ON DAVIS

Figure 10 shows optical-flow predictions on DAVIS without dataset-specific fine-tuning. The examples include articulated motion, thin structures, and small moving objects. They provide a qualitative illustration of the model's behavior on real-world video.