GAFlow: Incorporating Gaussian Attention into Optical Flow

Ao Luo[1], Fan Yang[2], Xin Li[2], Lang Nie[3], Chunyu Lin[3], Haoqiang Fan[1], and Shuaicheng Liu[4,1*]
[1]Megvii Technology [2]Group 42 [3]Beijing Jiaotong University
[4]University of Electronic Science and Technology of China

Abstract

Optical flow, or the estimation of motion fields from image sequences, is one of the fundamental problems in computer vision. Unlike most pixel-wise tasks that aim at achieving consistent representations of the same category, optical flow raises extra demands for obtaining local discrimination and smoothness, which yet is not fully explored by existing approaches. In this paper, we push Gaussian Attention (GA) into the optical flow models to accentuate local properties during representation learning and enforce the motion affinity during matching. Specifically, we introduce a novel Gaussian-Constrained Layer (GCL) which can be easily plugged into existing Transformer blocks to highlight the local neighborhood that contains fine-grained structural information. Moreover, for reliable motion analysis, we provide a new Gaussian-Guided Attention Module (GGAM) which not only inherits properties from Gaussian distribution to instinctively revolve around the neighbor fields of each point but also is empowered to put the emphasis on contextually related regions during matching. Our fully-equipped model, namely Gaussian Attention Flow network (GAFlow), naturally incorporates a series of novel Gaussian-based modules into the conventional optical flow framework for reliable motion analysis. Extensive experiments on standard optical flow datasets consistently demonstrate the exceptional performance of the proposed approach in terms of both generalization ability evaluation and online benchmark testing. Code is available at `https://github.com/LA30/GAFlow`.

1. Introduction

Optical flow aims to establish pixel-wise correspondences across images, playing a crucial role in video understanding. It unifies representation learning and feature matching as a problem of pixel-wise motion inference. Modern optical flow models typically focus on either improving representation learning techniques (*e.g.*, alternative learning [40, 20, 54] and reinforcement learning [1]) or refining feature similarity measurement methodologies (*e.g.*, 4D correlation volumes [42] or 4D Transformer [16]). Despite these significant advancements, a glaring limitation persists: these models largely neglect the exploration of local structural information. This oversight hinders their performance, particularly in challenging scenarios involving large motions, occlusions, blurring effects, and shifts in appearance. Such situations demand an elevated focus on local discrimination and flow consistency. This brings us to an intriguing inquiry: *Is it feasible to architect optical flow models that intrinsically focus on local structural information during both representation learning and feature matching?*

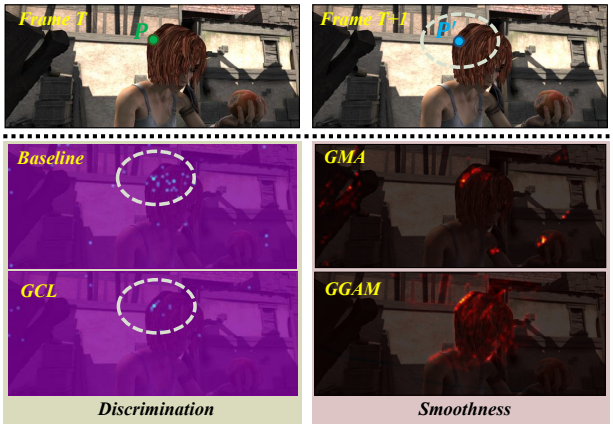


Figure 1: **Visualization** of feature discrimination (left) and smoothness constraint (right). For a random point P in frame t, our GCL enhances feature discrimination by emphasizing fine-grained structural details. Concurrently, our GGAM effectively captures smoothness constraints, centering on pertinent local regions.

In response to the aforementioned challenge, we present the Gaussian Attention Flow network (GAFlow), a pioneering framework that leverages Gaussian Attention (GA) to inform both the feature encoder and the matching mod-

*Corresponding author

ule. Central to our approach is a novel attention mechanism equipped with a learnable Gaussian kernel. This mechanism naturally prioritizes the local surroundings of each point, thereby accentuating the importance of local structural information. For the representation learning component, we introduce a novel Gaussian-Constrained Layer (GCL) embedded within the canonical feature encoder. This layer, fortified with a learnable Gaussian kernel, naturally prioritizes the local surroundings of each point, thus accentuating the significance of local structural data. It's worth noting that while the GCL builds upon the pixel-wise modeling typical of the Vision Transformer block, it possesses a sharper focus on local connections, which effectively enhances the feature discrimination for cross-frame matching, as depicted in Fig. 1. Moreover, our GCL is dynamic, comprising learnable parameters that can be fine-tuned in concert with the encoder.

In optical flow prediction, traditional techniques prioritize the matching function, viewing it as indispensable when paired with distinctive features. These methods incorporate feature-similarity measures and smoothness constraints to optimize results [15, 5, 6]. With the advent of deep learning, the spotlight has largely shifted to feature-similarity [16, 42], often sidelining the crucial smoothness constraint. When considering it, the implicit smoothness loss is primarily employed [51, 26], presuming uniform flow fields —– a simplification that overlooks intricate object deformations. While some approaches employ Graph techniques [30] or Transformers [22] to capture global motion, they tend to neglect the inherent locality of motion, potentially introducing inaccuracies. The challenge of leveraging neural networks to model motion relationships remains somewhat uncharted.

An ideal neural module for motion modeling should have three key properties: **i) Neighbourhood priority**. The motion of object(s) appears locally in visual scenes, and thus the module should instinctively focus more on the nearest neighbor fields for each pixel. **ii) Matching-prior awareness**. Previous work [4] shows that matching prior helps in large displacements and avoids over-smoothing; **iii) High-order relation centered**. Mining high-order relations [30, 22, 29] is essential for dealing with occlusions and lighting changes, as low-level similarities (like color) are often fragile. It would also be advantageous if this module's parameters could be trained in a data-driven manner.

Targeting the above goals, we introduce a novel Gaussian-Guided Attention Module (GGAM) to explicitly model the motion affinities for optical flow. To meet the neighborhood priority requirement, our GGAM is formulated as Gaussian-guided attention that naturally emphasizes the neighboring fields of each point. Second, unlike conventional Non-Local operation [44], our GGAM is built across the context feature map and embedded correlation (cost) volume for more comprehensive relation modeling. Specifically, for capturing the matching-prior knowledge, the 4D correlation volumes [42] are mapped to be the amplitudes and offsets. The amplitude for each position is set to the Gaussian attention for scaling its amplitude value and the offsets are encoded by the Gaussian attention via the warping operation. It enables free-form deformation of the Gaussian attention and adaptive attention for handling the large displacement of objects. For the last requirement, we draw inspiration from self-attention operations, and map the context feature to the query and key features for modeling the appearance self-similarities. All these operations are fully differentiable in our GGAM.

Our fully-equipped model, called Gaussian Attention Flow network (GAFlow), unites GCL and GGAM to conduct motion analysis by considering both feature similarities and motion affinities. It achieves the top performance on both Sintel and KITTI benchmarks with limited extra computational cost. Overall, the main **contributions** of this paper are: **1)** We introduce a novel approach to enhance the local properties of underlying representations. Our proposed Gaussian-Constrained Layer (GCL) can work complementarily with the standard feature encoder to build more discriminative features for optical flow. **2)** We analyze three important properties for motion affinity modeling, leading to a novel Gaussian-Guided Attention Module (GGAM). For the first time, we show that it is feasible to capture the local relations by learning the Gaussian attention and refining the motion fields for a more reliable optical flow estimation. **3)** We unite our GCL and GGAM into the contemporary optical flow architecture, making the model stronger at highlighting local structural information. Our Gaussian Attention Flow networks set new records on a variety of benchmarks, *e.g.,* Sintel (clean and final) and KITTI datasets, and outperform existing optical flow models by a relatively large margin.

2. Related Work

Optical Flow. In the epoch of deep learning, early optical flow models, such as [2, 46], viewed flow field prediction as a mapping challenge, translating two input frames into corresponding flow fields, predominantly harnessing the potency of data. Subsequent models significantly elevated flow estimation accuracy either by adopting more robust representation learning paradigms [40, 20, 54] or by implementing explicit feature-similarity measurement techniques [50, 43]. With the integration of advanced feature learning and cost-volume filtering modules, unsupervised methods have also witnessed remarkable advancements [31, 27, 25]. A notable development is RAFT [42], which employs a 4D correlation volume and a recurrent strategy for optical flow estimation. Building on its recurrent feature-matching paradigm, a slew of contempo-

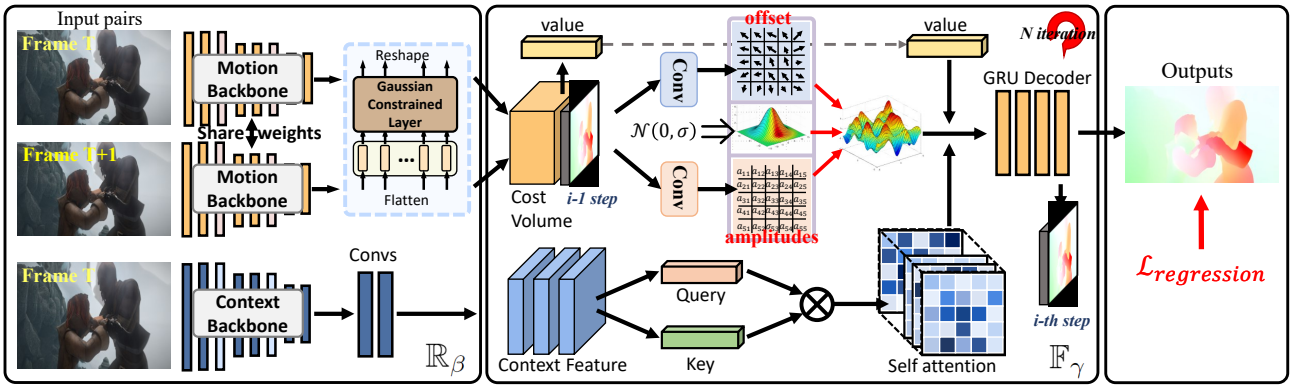


Figure 2: **An overview** of the proposed Gaussian Attention Flow network (GAFlow), architected with recurrent learning at its core. During the decoding process, the residual flows are iteratively refined and accumulated to derive the final flow field. "$\times$" denotes multiplication. Best viewed in color.

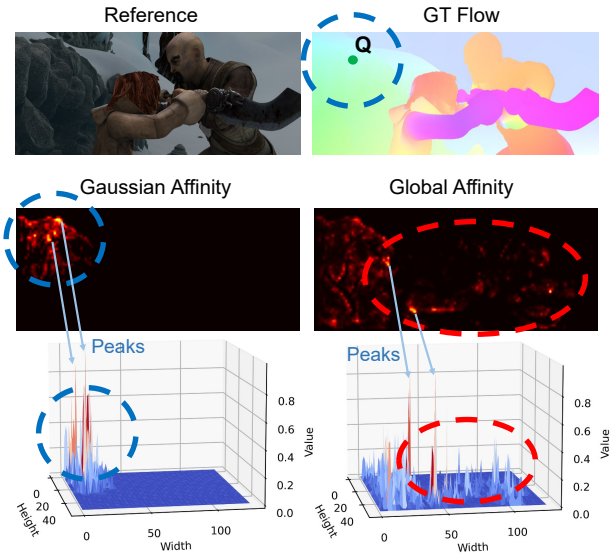


Figure 3: **A simple illustration** of the Gaussian constraint. "Q" represents the query point. The blue circle represents contextual affinity, mirroring the analogous patterns observed in the ground truth flow. Through the Gaussian constraint, we can effectively filter out deceptive relations, as highlighted within the red circle.

rary models [3, 13, 32, 21, 8] have substantially bolstered the dependability of optical flow predictions. To navigate challenges like occlusions and blur effects, innovative tactics—including joint representation learning [53], feature-driven flow regularization [19], iterative refinement [20, 18], and comprehensive motion analysis [22, 30]—have been employed. Addressing the smoothness constraint, [22] proposed a global motion refinement strategy anchored on attention mechanisms. Meanwhile, [30] captures motion affinities using graph techniques and [29] introduced kernel patch attention (KPA) for the same purpose, though KPA's static kernel window size could potentially curb its efficacy in managing variations in scale and shape.

Self-Attention. Self-attention operations have become instrumental in computer vision for discerning global contexts in images and videos [44, 45, 57]. Notably, non-local attention, exemplified by [57] and [17], is pivotal in visual Transformers [10, 12]. Recent advancements, such as Swin [28] and NAT [14], harness the *inductive bias* to enhance Transformers by applying attention within local windows. Inspired by these, our paper presents two Gaussian-based attention modules for optical flow.

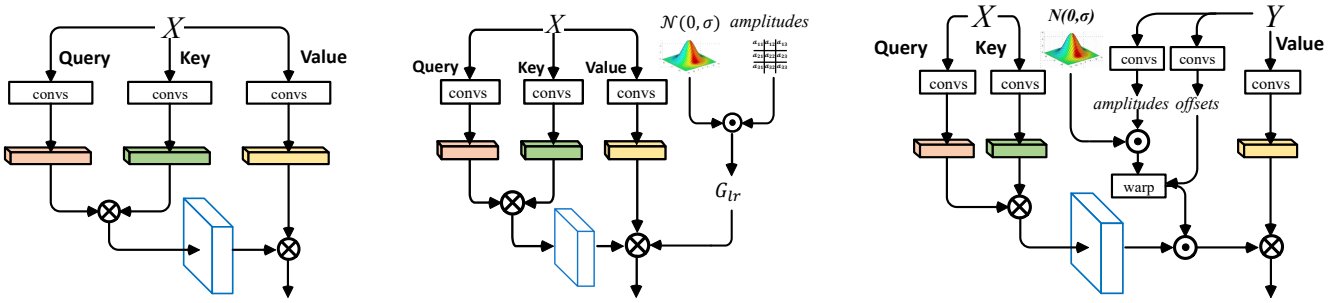
3. Our Approach

3.1. Preliminaries

Motivation. Fig. 3 visually elucidates the core motivation driving our research. It can be observed that the ground truth optical flow field reveals the intricate nature of motion affinities. These affinities are not only deeply intertwined with contextual relationships but also exhibit pronounced local characteristics. Conventional methods often fall short in aligning with the inherent dynamics of the optical flow field. This observation has propelled us to integrate Gaussian attention into the optical flow model, aiming for a marked enhancement in its precision.

3.2. Overview

Our GAFlow encompasses two pivotal sub-modules: *representation learning* module ($\mathbb{R}_\beta$) and *feature matching* module ($\mathbb{F}_\gamma$), as visualized in Fig. 2. Specifically, $\mathbb{R}_\beta$ represents the functions of acquiring the feature maps ($\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_c$),



(a) Non-local Attention (b) Gaussian Constrained Attention (c) Gaussian Guided Attention

Figure 4: **Architecture comparisons.** (a) Standard Non-local Attention, (b) Our Gaussian-Constrained Attention tailored for representation enhancement, and (c) Our Gaussian-Guided Attention, designed to model motion affinities using learnable Gaussian attention. Here, **X** represents the basic/context feature, while **Y** signifies the motion feature.

corresponding to the feature encoders combined with the proposed Gaussian-Constrained Layer (GCL). $\mathbb{F}_\gamma$ pertains to the recurrent flow decoder, fortified with our bespoke Gaussian-Guided Attention Module (GGAM).

At their core, these Gaussian-centric modules can be seamlessly integrated into flow models. Their inclusion accentuates local attributes during both the phases of representation learning and feature matching. Subsequent sections further unpack the mechanics of our GCL and GGAM.

3.3. Gaussian-Constrained Layer

The pixel-wise matching is susceptible to feature quality; Yet, the appearance changes caused by the motion blur and the illumination variations lower the feature discrimination. To avoid ambiguities, we propose Gaussian-Constrained Layer (GCL) to obtain the fine-grained structural information and locally discriminative representations for feature matching. Our GCL is designed based upon the standard Transformer [10, 28] block (see Fig. 4 (b)). Specifically, given the base feature $\mathbf{x}$, it is formulated as:

$$\begin{aligned} \hat{\mathbf{x}} &= \mathrm{GCA}(\mathrm{LN}(\mathbf{x})) + \mathbf{x}, \\ \mathbf{y} &= \mathrm{FFN}(\mathrm{LN}(\hat{\mathbf{x}})) + \hat{\mathbf{x}}, \end{aligned} \tag{1}$$

where $\mathrm{LN}(\cdot)$ and $\mathrm{FFN}(\cdot)$ denote the general layer normalization and feed-forward network respectively in Transformer block. $\mathrm{GCA}(\cdot)$ indicates Gaussian Constrained Attention, which is the core component in our GCL.

Here, we formulate GCA as a task-specific local operation, which not only avoids global misleading context but also reduces the complexity of attention computation. Therefore, it can be formulated as:

$$\begin{aligned} &\mathrm{Q}_i = L_i^Q(\mathbf{x}'), \mathrm{K}_i = L_i^K(\mathbf{x}'), \mathrm{V}_i = L_i^V(\mathbf{x}'), \\ &h_i = \mathcal{G}_\mathrm{A}(\mathrm{Q}_i, \bar{\mathrm{K}}_i, \bar{\mathrm{V}}_i), \\ &\mathrm{H} = \mathrm{Concat}(h_1, h_2, ..., h_n), \end{aligned} \tag{2}$$

where $\mathbf{x}' = \mathrm{LN}(\mathbf{x})$. $L_i^Q(\cdot)$, $L_i^K(\cdot)$, and $L_i^V(\cdot)$ denote linear projections for the i-th head. $\mathcal{G}_\mathrm{A}(\cdot)$ is Gaussian attention function, which takes the query feature Q_i, and the regional features of key $\bar{\mathrm{K}}_i$ and value $\bar{\mathrm{V}}_i$ as inputs:

$$\mathcal{G}_\mathrm{A}(\mathrm{Q}, \bar{\mathrm{K}}, \bar{\mathrm{V}}) = \mathrm{Softmax}(\mathrm{G}_{lr} \cdot (\mathrm{Q}\bar{\mathrm{K}}^\mathsf{T})/\sqrt{d}) \cdot \bar{\mathrm{V}}, \tag{3}$$

where G_{lr} means a learnable Gaussian kernel with the shape of $k \times k$. It is initialized as a standard Gaussian distribution and can be updated by adding a learnable amplitude matrix $\mathcal{A}$ during model training. In the inference process, G_{lr} works as a constraint mask to reorganize the weights of attentive feature aggregation.

Let the neighborhood of a pixel at the point p be $\mathcal{N}(p)$; thus the attention on a single pixel can be defined as:

$$h_{(p)} = \mathrm{Softmax}(\mathrm{G}_{lr} \cdot (\mathrm{Q}_{(p)}\bar{\mathrm{K}})^\mathsf{T}_{\mathcal{N}(p)}/\sqrt{d}) \cdot \bar{\mathrm{V}}_{\mathcal{N}(p)}. \tag{4}$$

Note that the operating range is scalable with the varying region of $\mathcal{N}(p)$. For instance, it can be extended to all pixels (*i.e.,* $\mathcal{N}(p)$ is equivalent to the image size), leading to a global self-attention in a Gaussian-constrained manner.

3.4. Gaussian-Guided Attention Module

Our Idea/Formulation. In image processing, the Gaussian filter is a widely-used operator that provides a linear scale space for smoothing, where the input image is smoothed at a constant rate in all directions [47]. Here, our idea is to apply the spatially variant Gaussian to smooth the motion feature $\mathbf{f}_m$. Formally, it can be given as:

$$(\mathrm{F} * \mathcal{G})(p) = \sum_{p_i \in \mathcal{N}_p} \mathcal{G}(p_i - p) {\mathbf{f}_m}^i, \tag{5}$$

where $\mathcal{N}_p$ denotes a square neighborhood, centered at the pixel p, with the pre-defined kernel size $k \times k$, and p_i indicates the position i in 2D grid space within $\mathcal{N}_p$. ${\mathbf{f}_m}^i$ refers to the feature at position i within $\mathbf{f}_m$, utilized in the calculation. $\mathcal{G}(\cdot)$ is a 2D Gaussian function, which is defined as:

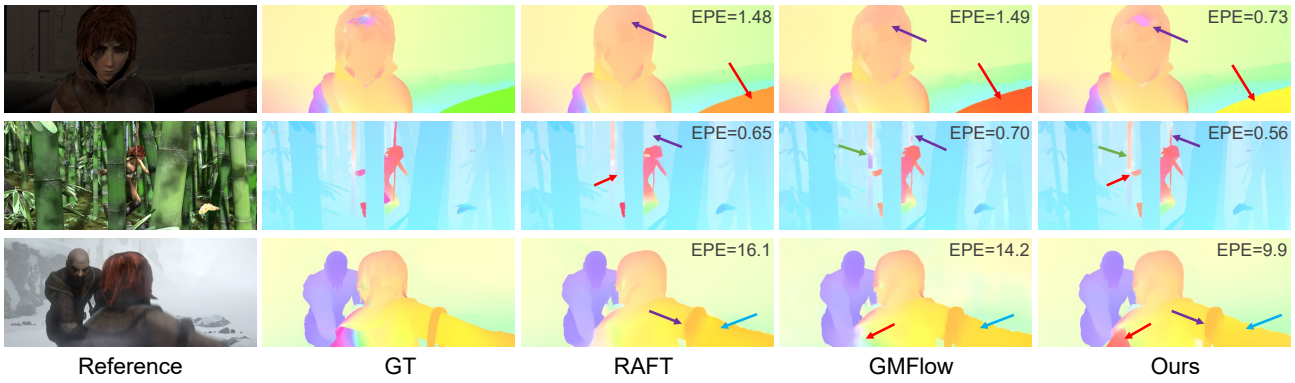


Figure 5: **Qualitative comparisons** with RAFT [42] and GMFlow [49] on some challenging samples, including large motion, occlusion and motion blur, of Sintel `test` set. The quantitative results are provided by the official website.

$$\mathcal{G}(x,y) = A\exp(-(\frac{(x-x_0)^2}{2\sigma_x^2} + \frac{(y-y_0)^2}{2\sigma_y^2})), \quad (6)$$

where A denotes the static amplitude ($A = 1$ as the default setting) and $p_0 = (x_0, y_0)$ means the center point. $(\sigma_x; \sigma_y)$ indicate the variances which are set to $\sigma_x = \sigma_y = \sigma$, determining the operating range of smoothness. However, directly applying the Gaussian smoothing is sub-optimal for capturing the local properties, as the kernel is rigid (or fixed) and the clutters could be inevitably involved in providing false guidance. Next, we show how to make the Gaussian attention more adaptive and also end-to-end trainable.

Gaussian-Guided Attention with Context (GGAC). To begin with, we incorporate the contextual information, which preserves the fine-grained structural information, into the Gaussian Attention. Specifically, given the context feature $\mathbf{f}_c \in \mathbb{R}^{c\times h\times w}$ and motion feature $\mathbf{f}_m \in \mathbb{R}^{c\times h\times w}$, we rewrite Eq. (5) as:

$$(\mathrm{F} * \mathcal{G}_{\mathtt{C}})(p) = \sum_{p_i\in\mathcal{N}_p} \mathcal{G}_{\mathtt{C}(p_i-p)}(\mathbf{f}_c)\rho(\mathbf{f}_m)^i, \quad (7)$$

where $\mathcal{G}_{\mathtt{C}(p_i-p)}(\cdot)$ is an adaptive kernel function taking the context features in $\mathcal{N}_p$ as inputs to obtain the context-preserving kernel. $\rho(\cdot)$ is a linear operation for producing the embedded motion feature. We formulate the context-preserving kernel function as:

$$\mathcal{G}_{\mathtt{C}(p_i-p)}(\mathbf{f}_c) = \mathcal{G}'(p_i - p)\mathcal{F}(\mathbf{f}_c), \quad (8)$$

where $\mathcal{G}'(p_i-p) \in \mathbb{R}^{k\times k\times h\times w}$ denotes the expanded Gaussian kernel in the spatial dimension, which is prepared for generating the data-dependent kernel matrics with the shape of $k \times k$ based on the corresponding weights on each position in $h \times w$, and $\mathcal{F}(\cdot)$ is the dynamic weight function for mining the guidance prior on the context feature $\mathbf{f}_c$.

Rather than simply utilizing a learnable matrix as in convolution kernels, we follow [22] to apply the embedded Gaussian with normalization [44] on the context feature $\mathbf{f}_c$ to produce the attention map. To match the dimension of the pre-defined Gaussian kernel $\mathcal{G}'(p_i - p) \in \mathbb{R}^{K\times N}$, where $N = h \times w$, and $K = k \times k$ denoting the spatial dimension of kernel size, a naive solution is to discard the redundant information in $\mathcal{F}(\mathbf{f}_c)$. Here, we present a kernel-based context-preserving attention operation, which takes the asymmetric query and key feature maps in the spatial dimension as inputs. In detail, given the embedded context features $\theta(\mathbf{f}_c) \in \mathbb{R}^{c\times h\times w}$, we employ an unfolding operator (*e.g.*, torch.nn.Unfold) to get $\mathrm{U}(\theta(\mathbf{f}_c))$, thus the dimension is transformed from $c \times N$ to $c \times N \times K$. Then we measure the similarities between all pairs within the kernel region as:

$$\mathcal{F}(\mathbf{f}_c) = \frac{\exp(\mathrm{U}(\theta(\mathbf{f}_c))_l{}^T \phi(\mathbf{f}_c)_j/\sqrt{c})}{\sum_{\forall j}\exp(\mathrm{U}(\theta(\mathbf{f}_c))_l{}^T \phi(\mathbf{f}_c)_j/\sqrt{c})}, \quad (9)$$

where $l \in K$ and $j \in c$, and the size of $\mathcal{F}(\mathbf{f}_c)$ is $K \times N$. $\theta(\cdot)$ and $\phi(\cdot)$ indicates linear projections for channel-wise feature aggregation [22].

Gaussian-Guided Attention with Deformation (GGAD). While Gaussian attention with context effectively addresses the problem of flow field over-smoothing caused by the data-agnostic Gaussian kernel, it still faces a limitation due to the rigid operating region of the Gaussian kernel. This constraint hinders the smoothness performance for motion feature refinement. To mitigate this issue, we take a further step to optimize the operating region of the Gaussian kernel by learning the deformable kernels in a data-driven manner, as shown in Fig. 4 (c). To avoid the static amplitude A of the Gaussian function, we design a dynamic amplitude operator based on the cross-frame matching features. In practice, similar to Eq. (7), our Gaussian-guided attention can be further formulated as:

$$(\mathrm{F} * \mathcal{G}_{\mathtt{D}})(p) = \sum_{p_i\in\mathcal{N}_p} \mathcal{G}_{\mathtt{D}(p_i-p)}(\mathbf{f}_c, \mathbf{f}_m)\rho(\mathbf{f}_m), \quad (10)$$

Training	Method	Reference	Sintel (Val)		KITTI-15 (Val)		Sintel (Test)		KITTI-15 (Test)
			Clean	Final	EPE	F1-all	Clean	Final	F1-all
Val: C + T / Test: +S+K+H	RAFT [42]	ECCV-20	1.43	2.71	5.04	17.4	1.61	2.86	5.10
	SCV [23]	CVPR-21	1.29	2.95	6.80	19.3	1.77	3.88	6.17
	GMA [22]	ICCV-21	1.30	2.74	4.69	17.1	1.39	2.47	5.15
	SeparableFlow[52]	ICCV-21	1.30	2.59	4.60	15.9	1.50	2.67	4.64
	Flow1D [48]	ICCV-21	1.98	3.27	6.69	23.0	2.24	3.81	6.27
	AGFlow [30]	AAAI-22	1.31	2.69	4.82	17.0	1.43	2.47	4.89
	GMFlowNet [55]	CVPR-22	1.14	2.71	4.24	15.4	1.39	2.65	4.79
	GMFlow [49]	CVPR-22	1.08	2.48	7.77	23.4	1.74	2.90	9.32
	CRAFT [37]	CVPR-22	1.27	2.79	4.88	17.5	1.45	2.42	4.79
	DIP [56]	CVPR-22	1.30	2.82	4.29	**13.7**	1.44	2.83	**4.21**
	KPA-Flow [30]	CVPR-22	1.28	2.68	4.46	15.9	1.35	2.36	4.60
	OCTC [21]	CVPR-22	1.31	2.67	4.72	16.3	1.41	2.57	4.33
	SKFlow [41]	NeurIPS-22	1.22	2.46	4.27	15.5	1.28	2.27	4.84
	GAFlow (ours)		**1.02**	**2.45**	**3.98**	15.0	**1.21**	**2.24**	4.52
A / +TSKHV	RAFT-A [39]	CVPR-21	1.95	2.57	4.23	-	2.01	3.14	4.78
	RAFT-it [38]	ECCV-22	1.74	2.41	4.18	**13.4**	1.55	2.90	**4.31**
I+CT / +SKH	FlowFormer [16]	ECCV-22	1.01	2.40	4.09	14.7	1.16	2.09	4.68
	GAFlow-FF (ours)		**0.95**	**2.34**	**3.92**	13.9	**1.15**	**2.05**	4.42

Table 1: **Quantitative comparison** with state-of-the-art models on standard benchmarks for cross-dataset evaluation and online testing. "C+T" indicates the models trained on FlyingChairs and FlyingThings for generalization ability evaluation. "+S+K+H" denotes more training data involved from Sintel, KITTI-2015, and HD1K. "A" indicates the models are trained on AutoFlow [39] dataset for Val set evaluation. "V" denotes VIPER [36] dataset. "I" means the encoders are pre-trained on ImageNet [9]. The best and second-best results are marked in **bold** and underline, respectively.

where $\mathcal{G}_{\mathtt{D}(p_i-p)}(\cdot)$ is the deformable Gaussian kernel function that is used to produce the more flexible Gaussian attention for adaptively capturing the relevant information and avoiding rigid kernel boundary in a data-dependent manner. Specifically, the deformable Gaussian kernel is given by:

$$\mathcal{G}_{\mathtt{D}(p_i-p)}(\mathbf{f}_c, \mathbf{f}_m) = \mathcal{A}(\mathbf{f}_m)\mathcal{D}_{(p_i-p)}(\mathbf{f}_m)\mathcal{F}(\mathbf{f}_c), \tag{11}$$

where $\mathcal{A}(\cdot)$ indicates the amplitude operator taking the motion feature $\mathbf{f_m}$ to produce adaptive amplitudes, $\mathcal{F}(\mathbf{f}_c)$ is the kernel-based context-preserving attention matrix as in Eq. (9), and $\mathcal{D}_{(p_i-p)}(\cdot)$ denotes the deformable Gaussian operator, which can be formulated as:

$$\mathcal{D}_{(p_i-p)}(\mathbf{f}_m) = \mathcal{W}(\mathcal{G}'(p_i-p), \mathcal{O}(\mathbf{f}_m)), \tag{12}$$

where $\mathcal{G}'(p_i-p) \in \mathbb{R}^{k\times k\times h\times w}$ indicates the spatially expanded Gaussian kernel as in Eq. (8), and $\mathcal{O}(\mathbf{f}_m)$ denotes the offset maps predicted from the motion feature. $\mathcal{W}(\cdot)$ is the warp function taking the obtained offsets as inputs and operating on the Gaussian kernel.

Another important component in the deformable Gaussian kernel (Eq. (11)) is the amplitude operator $\mathcal{A}(\cdot)$, which is given by:

$$\mathcal{A}(\mathbf{f}_m) = 1 + \vartheta(\mathbf{f}_m)\lambda, \tag{13}$$

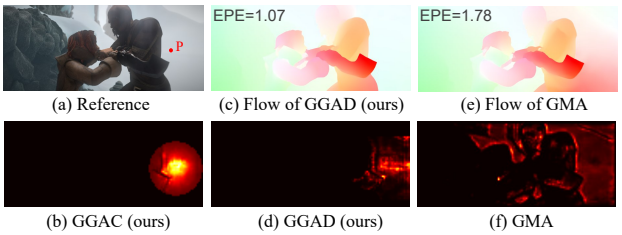
where λ indicates a learnable parameter to constrain the fluctuation of the generated amplitude, and $\vartheta(\cdot)$ is a linear function. Finally, similar to the expanding strategy in the deformable Gaussian operator, the learned amplitude $\mathcal{A}(\mathbf{f}_m)$ can be easily transformed to operate on the Gaussian kernel $\mathcal{G}'(p_i-p)$ for generating a more flexible one.

4. Experimental Results

4.1. Implementation Details

For fair comparison with existing methods [22, 37, 29], we first plug 1 GCL and 1 GGAM into RAFT [42] to build a small model, *i.e.,* GAFlow-S. Specifically, Gaussian-Constrained Layer (GCL) is appended to the motion encoder with $\sigma = 9$. Gaussian-Guided Attention Module (GGAM) performs feature smoothing on context and motion features, where we set sigma to 20 and 15 for Sintel and KITTI, respectively. Besides, to further explore the potential of our approach, we also employ a stronger baseline with POLA [55] modules and SKBlocks [41], which is regarded as the default model of GAFlow. Furthermore, we enhance the model's capabilities by integrating the proposed module into FlowFormer [16], resulting in a robust and advanced model termed GAFlow-FF.

All experiments are conducted based on PyTorch toolbox. During training our GAFlow model, we follow RAFT to set the batch sizes to 6 and adopt AdamW optimizer with one-cycle learning rate policy [42]. Similar to previous works [42, 22, 30], the synthetic datasets are also involved



(a) Reference (c) Flow of GGAD (ours) (e) Flow of GMA

(b) GGAC (ours) (d) GGAD (ours) (f) GMA

Figure 6: **A challenging sample** from the Sintel final pass. The second row provides attention maps of our GGAC, GGAD and GMA [22], respectively. "P" indicates the query point to obtain attention maps.

in model pre-training. The pre-training iterations on FlyingChairs [11] and FlyingThings [33] are set to 120k and 160k, respectively. Then the fine-tuning for Sintel online evaluation is conducted on the combined training data of Sintel [7], KITTI-2015 [34], and HD1K [24] for 160k iterations. Finally, the trained model requires another 50k iterations of finetuning on data of KITTI-2015 [34] before KITTI testing. We use only a single GPU and set the batch size to 1 for evaluation and online testing.

4.2. Comparison with State-of-the-Arts

Results on Sintel. We first compare the proposed approach with state-of-the-art methods for generalization evaluation on Sintel. As shown in `Val` sets of Tab. 1, our approach consistently achieves the best performance on all metrics. Specifically, GAFlow achieves the best EPE scores at 1.02 and 2.45 on the "C + T" setting. Besides, GAFlow-FF improves the SOTA performance to 0.95 and 2.34 in EPE on Sintel clean and final passes, respectively. For online testing, we follow prior works [22, 29, 37] to utilize the warm-start strategy [42]. GAFlow achieves the best EPE scores at 1.21 and 2.24 on the standard training setting. Moreover, GAFlow-FF sets a new record at 1.15 and 2.05 in EPE, outperforming recent works by a relatively large margin.

We compare the proposed GAFlow with the well-known methods RAFT [42] and GMFlow [49] on Sintel `test` set, and some qualitative comparisons are illustrated in Fig. 5. The results demonstrate that the proposed approach can effectively leverage the local structural information and high-order relations to resolve the ambiguities in motion learning, leading to a more robust and flexible model for handling the challenges in optical flow estimation.

Results on KITTI. As shown in Tab. 1, GAFlow achieves 3.98 in EPE and 15.0% in F1-all on the KITTI `Val` set, which is comparable with the SOTA model FlowFormer yet with less computational overhead. Moreover, GAFlow-FF further boosts the scores to 3.92 and 13.9%. For online testing, our models achieve the top-ranked performances at 4.52% and 4.42% in F1-score.

Sintel	Type	RAFT[42] (EPE)	GMA[22] (EPE)	KPA[29] (EPE)	**GAFlow-S** (EPE)
Clean (train)	s0-10	0.32	0.27	0.26	**0.24**
	*s*10-40	1.55	1.35	1.38	**1.29**
	*s*40+	8.83	8.55	8.43	**7.92**
	All	1.41	1.31	1.28	**1.21**
Final (train)	*s*0-10	0.52	0.54	0.54	**0.51**
	*s*10-40	3.00	3.07	3.02	**2.83**
	*s*40+	17.45	17.57	17.34	**17.02**
	All	2.69	2.74	2.68	**2.56**

Table 2: **Quantitative results** on different displacements. All methods are trained on "C+T" and evaluated on Sintel for fair comparison. *s*0-10, *s*10-40 and *s*40+ denote the ground truth flow value belonging to 0-10, 10-40 and 40+ pixels [7], respectively.

4.3. Ablation Analysis

Comparison with GMA. GMA [22] presents a global strategy for handling occlusion in flow estimation. As shown in Tab. 2, our GAflow-S achieves better performance on all metrics. This is because the global attention on context features involves too much misleading information for motion guidance, which contains long-range similarities in appearance but is improper for motion refinement (see Fig. 6 (f)). In contrast, our approach effectively takes the local property of regional context affinities into consideration, and leverages the extracted high-order relations with the flexible operating range to better infer the flow fields.

Comparison with KPA. The recent work KPA-Flow [29] shares a similar insight with our approach, leveraging local attention to avoid global misleading information. However, its intrinsic defects in formulation inevitably hinder the effect of motion refinement. First, the zero padding patches (out of the image) lead to obvious deviation for similarity measurement. Second, the neighboring pixels could be partitioned into distinct pre-defined (fixed) patch and kernel windows, leading to abrupt shifts in operating regions. In contrast, our approach enables a flexible constraint and freeform operating regions in a learnable manner, ensuring the proper and stable guidance pattern. Moreover, we provide quantitative comparisons in Tab. 2, where our approach achieves better performance on all metrics.

Ablation for Feature Enhancement. In Tab. 3 (# 1), we first compare the proposed GCL with the widely recognized Transformer blocks [28, 55, 35], which have been extensively demonstrated to be effective for feature extraction. Swin Transformer [28] requires at least two blocks to perform the window shifting strategy, consuming more computation overhead. For other methods, only one block is applied for fair comparison. POLA [55] has been shown to be an effective local attention-based approach for fea-

Method	Sintel (Val)		KITTI-15 (Val)		Param.
	Clean	Final	EPE	F1-all	
RAFT [42]	1.43	2.71	5.04	17.4	5.3M
# 1: Feature Enhancement					
Swin Trans. [28]	1.45	2.75	5.16	17.3	7.3M
POLA$_{\times 1}$ [55]	1.42	2.68	4.91	16.9	6.5M
cosFormer [35]	1.29	2.82	4.95	17.3	6.7M
L.Kernel	1.42	2.76	4.69	16.6	6.5M
GCL	1.33	2.67	4.61	16.5	6.5M
# 2: Stronger Encoder					
POLAs [55]	1.18	2.63	4.52	16.6	10.0M
No	1.33	2.67	4.61	16.5	6.5M
# 3: Smoothing Pattern					
+ GGAC	1.13	2.62	4.33	15.8	10.1M
+ GGAD	1.08	2.56	4.26	15.6	10.1M
# 4: Compatibility with SOTA modules					
Refine.Sp4 [49]	1.03	2.45	4.10	14.6	12.5M
SKBlocks [41]	1.02	2.45	3.98	15.0	10.4M

Table 3: **Ablation experiments.** L.Kernel indicates the learnable kernel without Gaussian. Settings as default are underlined. Refer to Sec. 4.3 for detailed comparisons.

ture extraction with 6 or 12 stacked blocks (requiring extra $4.35/8.7$ M parameters). As shown in the table, using one block (POLA$_{\times 1}$) significantly weaken the power. The advanced linear Transformer cosFormer [35] exhibits a lower EPE score on Sintel Clean, whereas our GCL performs better on other metrics. Besides, we conduct an extra ablation for using a learnable kernel without Gaussian (L.Kernel). As can be seen, our GCL outperforms L.Kernel across all metrics. Particularly on the challenging Sintel, without a Gaussian constraint, performance declines significantly due to the uncontrolled misleading similarities at the kernel border. All experiments illustrate that, requiring similar computational overhead, the proposed GCL surpasses other methods by a relatively large margin.

Ablation for Stronger Encoder. The smoothness constraint for flow estimation can be naturally regarded as a complementary component to the feature matching process, and benefits from stronger encoders for feature extraction. Thus, to further explore the potential of our approach, we follow prior work [55] to employ a stronger motion encoder with the standard POLA blocks, which largely boosts the performance on Sintel clean pass, as in Tab. 3 (# 2). Inspired by GMFlowNet [55], we build two types of encoding modules for separately handling Sintel and KITTI datasets.

Ablation for Smoothing Pattern. The proposed GGAC is capable of producing flexible weight matrices for motion feature smoothing, which effectively tackles the issue of flow field over-smoothing caused by the data-agnostic Gaussian kernel. However, the issues of rigid operating region and static amplitude still affect the motion refinement (see Fig. 6 (b) and Tab. 3 (# 3)). Thus we further improve the approach with the spatially deformable kernel with the dynamic amplitude in a data-driven manner, *i.e*., GGAD, which boosts the performance by around 3%.

Method	KITTI-15 (Val)		Param.	Time (s)
	EPE	F1-all		
GMFlowNet [55]	4.24	15.4	9.3M	0.32
FlowFormer [16]	4.09	14.7	18.2M	1.77
GAFlow (ours)	3.98	15.0	10.4M	0.48
GAFlow-FF (ours)	3.92	13.9	18.2M	1.85

Table 4: Quantitative comparisons for computational complexity. The input size is 376×1248 as in KITTI dataset.

Ablation for Compatibility with SOTA modules. In Tab. 3 (# 4), similar to prior work [49], we further perform flow refinement with the flow decoder on 1/4 features, termed Refine.Sp4, which helps to boost the performance on both Sintel and KITTI datasets. Moreover, we follow SKFlow [41] to employ SKBlocks in the motion encoding and state updating modules of the flow decoder. The ablation indicates that the GGAD module is compatible with SKBlocks, and the integration of the decoder leads to improved performance across all metrics, while introducing minimal additional parameters ($10.1 \rightarrow 10.4$ M).

Runtime Comparison. The comparisons of KITTI evaluation results, parameters and runtime are presented in Tab. 4. Although FlowFormer [16] uses ImageNet for encoder pretraining, our GAFlow achieves a competitive performance (even better in online testing, see Tab. 1) while consuming fewer parameters by 7.8 M and reducing the inference time by 72.9%. Moreover, built on the previous SOTA model FlowFormer, our GAFlow-FF largely reduces the error by 4.8% with negligible computational complexity.

5. Conclusion

In this work, we introduce a novel Gaussian Attention Flow network (GAFlow) to explicitly highlight local properties during representation learning and enforce the motion affinity during matching. We deliver two Gaussian-based modules, *i.e*. Gaussian-Constrained Layer (GCL) and Gaussian-Guided Attention Module (GGAM), which works well with existing flow architectures and can greatly enhance the reliability of optical flow estimation. Extensive quantitative and qualitative comparisons performed on commonly-used datasets demonstrate that our method significantly outperforms the current alternatives.

Acknowledgements. This work is supported by Sichuan Science and Technology Program of China under grant No.2023NSFSC0462.

References

[1] Artemij Amiranashvili, Alexey Dosovitskiy, Vladlen Koltun, and Thomas Brox. Motion perception in reinforcement learning with dynamic objects. In *CRL*, 2018.

[2] Min Bai, Wenjie Luo, Kaustav Kundu, and Raquel Urtasun. Exploiting semantic information and deep matching for optical flow. In *ECCV*, 2016.

[3] Shaojie Bai, Zhengyang Geng, Yash Savani, and J Zico Kolter. Deep equilibrium optical flow estimation. In *CVPR*, 2022.

[4] Thomas Brox, Christoph Bregler, and Jitendra Malik. Large displacement optical flow. In *CVPR*, 2009.

[5] T. Brox, Andrés Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004.

[6] Andrés Bruhn, J. Weickert, and C. Schnörr. Lucas/kanade meets horn/schunck: combining local and global optic flow methods. *IJCV*, 2005.

[7] Daniel Butler, Jonas Wulff, Garrett Stanley, and Michael Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012.

[8] Changxing Deng, Ao Luo, Haibin Huang, Shaodan Ma, Jiangyu Liu, and Shuaicheng Liu. Explicit motion disentangling for efficient optical flow estimation. In *ICCV*, 2023.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[11] A. Dosovitskiy, P. Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015.

[12] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *TPAMI*, 2022.

[13] Yunhui Han, Kunming Luo, Ao Luo, Jiangyu Liu, Haoqiang Fan, Guiming Luo, and Shuaicheng Liu. Realflow: Em-based realistic optical flow dataset generation from videos. In *ECCV*, 2022.

[14] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *CVPR*, 2023.

[15] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 1981.

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

[17] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *ICCV*, 2019.

[18] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *CVPR*, 2018.

[19] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. A lightweight optical flow cnn—revisiting data fidelity and regularization. *TPAMI*, 2020.

[20] Junhwa Hur and S. Roth. Iterative residual refinement for joint optical flow and occlusion estimation. In *CVPR*, 2019.

[21] Jisoo Jeong, Jamie Menjay Lin, Fatih Porikli, and Nojun Kwak. Imposing consistency for optical flow estimation. In *CVPR*, 2022.

[22] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *ICCV*, 2021.

[23] Shihao Jiang, Yao Lu, Hongdong Li, and R. Hartley. Learning optical flow from a few matches. In *CVPR*, 2021.

[24] D. Kondermann, Rahul Nair, Katrin Honauer, Karsten Krispin, Jonas Andrulis, Alexander Brock, Burkhard Güssefeld, Mohsen Rahimimoghaddam, Sabine Hofmann, C. Brenner, and B. Jähne. The hci benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving. In *CVPRW*, 2016.

[25] Haipeng Li, Kunming Luo, Bing Zeng, and Shuaicheng Liu. Gyroflow+: Gyroscope-guided unsupervised deep homography and optical flow learning. *arXiv:2301.10018*, 2023.

[26] Liang Liu, Jiangning Zhang, Ruifei He, Yong Liu, Yabiao Wang, Ying Tai, Donghao Luo, Chengjie Wang, Jilin Li, and Feiyue Huang. Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In *CVPR*, 2020.

[27] Shuaicheng Liu, Kunming Luo, Ao Luo, Chuan Wang, Fanman Meng, and Bing Zeng. Asflow: Unsupervised optical flow learning with adaptive pyramid sampling. *TCSVT*, 2021.

[28] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[29] Ao Luo, Fan Yang, Xin Li, and Shuaicheng Liu. Learning optical flow with kernel patch attention. In *CVPR*, 2022.

[30] Ao Luo, Fan Yang, Kunming Luo, Xin Li, Haoqiang Fan, and Shuaicheng Liu. Learning optical flow with adaptive graph reasoning. In *AAAI*, 2022.

[31] Kunming Luo, Chuan Wang, Shuaicheng Liu, Haoqiang Fan, Jue Wang, and Jian Sun. Upflow: Upsampling pyramid for unsupervised optical flow learning. In *CVPR*, 2021.

[32] Xinglong Luo, Kunming Luo, Ao Luo, Zhengning Wang, Ping Tan, and Shuaicheng Liu. Learning optical flow from event camera with rendered dataset. In *ICCV*, 2023.

[33] N. Mayer, Eddy Ilg, Philip Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016.

[34] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015.

[35] Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran

Zhong. cosformer: Rethinking softmax in attention. In *ICLR*, 2022.

[36] Stephan R Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *ICCV*, 2017.

[37] Xiuchao Sui, Shaohua Li, Xue Geng, Yan Wu, Xinxing Xu, Yong Liu, Rick Goh, and Hongyuan Zhu. Craft: Cross-attentional flow transformer for robust optical flow. In *CVPR*, 2022.

[38] Deqing Sun, Charles Herrmann, Fitsum Reda, Michael Rubinstein, David J Fleet, and William T Freeman. Disentangling architecture and training for optical flow. In *ECCV*, 2022.

[39] Deqing Sun, Daniel Vlasic, Charles Herrmann, Varun Jampani, Michael Krainin, Huiwen Chang, Ramin Zabih, William T Freeman, and Ce Liu. Autoflow: Learning a better training set for optical flow. In *CVPR*, 2021.

[40] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and J. Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018.

[41] Shangkun Sun, Yuanqi Chen, Yu Zhu, Guodong Guo, and Ge Li. Skflow: Learning optical flow with super kernels. In *NeurIPS*, 2022.

[42] Zachary Teed and Jun Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020.

[43] Jianyuan Wang, Yiran Zhong, Yuchao Dai, K. Zhang, Pan Ji, and Hongdong Li. Displacement-invariant matching cost learning for accurate optical flow estimation. In *NeurIPS*, 2020.

[44] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[45] Zhengyang Wang, Na Zou, Dinggang Shen, and Shuiwang Ji. Non-local u-nets for biomedical image segmentation. In *AAAI*, 2020.

[46] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In *ICCV*, 2013.

[47] Andrew Witkin. Scale-space filtering: A new approach to multi-scale description. In *ICASSP*, 1984.

[48] Haofei Xu, Jiaolong Yang, Jianfei Cai, Juyong Zhang, and Xin Tong. High-resolution optical flow from 1d attention and correlation. In *ICCV*, 2021.

[49] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *CVPR*, 2022.

[50] Gengshan Yang and D. Ramanan. Volumetric correspondence networks for optical flow. In *NeurIPS*, 2019.

[51] Jason J Yu, Adam W Harley, and Konstantinos G Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *ECCV*, 2016.

[52] Feihu Zhang, Oliver J Woodford, Victor Adrian Prisacariu, and Philip HS Torr. Separable flow: Learning motion cost volumes for optical flow estimation. In *ICCV*, 2021.

[53] Chengqian Zhao, Cheng Feng, Dengwang Li, and Shuo Li. Of-msrn: optical flow-auxiliary multi-task regression network for direct quantitative measurement, segmentation and motion estimation. In *AAAI*, 2020.

[54] Shengyu Zhao, Yilun Sheng, Yue Dong, E. Chang, and Yan Xu. Maskflownet: Asymmetric feature matching with learnable occlusion mask. In *CVPR*, 2020.

[55] Shiyu Zhao, Long Zhao, Zhixing Zhang, Enyu Zhou, and Dimitris Metaxas. Global matching with overlapping attention for optical flow estimation. In *CVPR*, 2022.

[56] Zihua Zheng, Ni Nie, Zhi Ling, Pengfei Xiong, Jiangyu Liu, Hao Wang, and Jiankun Li. Dip: Deep inverse patchmatch for high-resolution optical flow. In *CVPR*, 2022.

[57] Zhen Zhu, Mengde Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. In *CVPR*, 2019.