Learning Optical Flow with Adaptive Graph Reasoning

Ao Luo[1], **Fan Yang**[2], **Kunming Luo**[1], **Xin Li**[2], **Haoqiang Fan**[1], **Shuaicheng Liu**[3,1]*

[1]Megvii Technology [2]Group 42
[3]University of Electronic Science and Technology of China

Abstract

Estimating per-pixel motion between video frames, known as optical flow, is a long-standing problem in video understanding and analysis. Most contemporary optical flow techniques largely focus on addressing the cross-image matching with feature similarity, with few methods considering how to explicitly reason over the given scene for achieving a holistic motion understanding. In this work, taking a fresh perspective, we introduce a novel graph-based approach, called adaptive graph reasoning for optical flow (AGFlow), to emphasize the value of scene/context information in optical flow. Our key idea is to decouple the context reasoning from the matching procedure, and exploit scene information to effectively assist motion estimation by *learning to reason* over the adaptive graph. The proposed AGFlow can effectively exploit the context information and incorporate it within the matching procedure, producing more robust and accurate results. On both Sintel clean and final passes, our AGFlow achieves the best accuracy with EPE of 1.43 and 2.47 pixels, outperforming state-of-the-art approaches by 11.2% and 13.6%, respectively. Codes will be publicly available at https://github.com/LA30/AGFlow.

Introduction

Optical flow is a fundamental task in video understanding and analysis, aiming to estimate the pixel-wise correspondence between two video frames. It has drawn continuous attention from both academia and industry due to its wide applications, *e.g.,* person-identification (**?**), visual tracking (**?**) and video inpainting (**?**). Recent years have witnessed significant breakthroughs made to push its performance frontier (**????**), but it remains challenging due to inherent ambiguity in textures, large displacements, occlusions, motion blur, and non-Lambertian effects.

Traditional optical flow algorithms formulate the dense matching as an energy minimization problem based on feature constancy and spatial smoothness (**???**). However, because the hand-designing features and optimization objectives are difficult to cover all scenarios, these approaches are not robust enough to deal with complex motions. As a powerful alternative to traditional methods, deep learning based approaches take the research of optical flow into a new level. Current optical flow methods have made great advances in: **i)** developing powerful data-driven, end-to-end learning paradigms (**??**); **ii)** designing multiple refinement strategies (**???**); **iii)** exploiting auxiliary information from related tasks (**?**); and **iv)** modeling pixel-wise relations for all pairs (**??**). Although these deep learning based approaches have shown the strong capability of matching across frames, they are subject to a significant limitation: current methods largely focus on addressing the matching similarity between features, lacking a holistic motion understanding of the given scene. Thus, the ambiguities (local variations) caused by motion blur, occlusion and large motions severely degrade the accuracy of current models (see Fig. 1), hindering their applications in the real world.

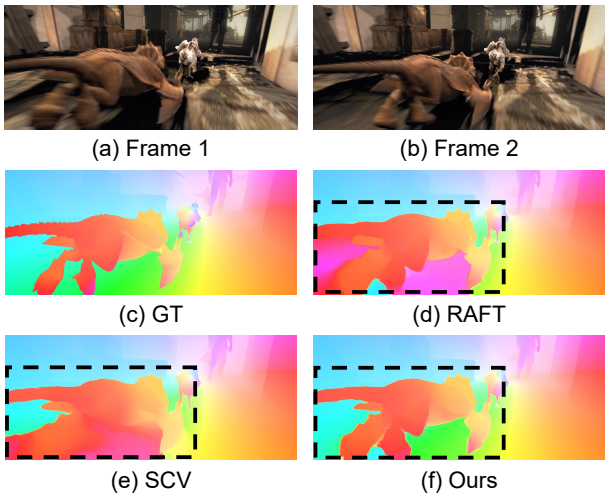

Figure 1: A challenging image pair with heavy motion blur from the final pass of Sintel `test` set. Unlike previous state-of-the-art methods (RAFT (**?**) and SCV (**?**)) suffering from ambiguous matching, our AGFlow is able to perform global reasoning conditioned on scene context for achieving a holistic motion understanding.

Taking a closer look at current optical flow methods (**??**), their success is majorly attributed to an important compo-

*Corresponding author

nent, namely 4D correlation volume, that typically model the correlations between features across frames. They expect the correlations achieved by using deep trainable features to surmount all ambiguities. However, evidence from (**?**) indicate that such correlations are vulnerable to variable feature representations on challenging conditions. To better aggregate context information within motion boundaries, existing arts (**??**) inject image features by concatenation operation, and encode scene information by using extra convolutional layers. But despite these attempts, the holistic motion understanding is far from being achieved: **i)** existing approaches capture scene information with only naive operations, *e.g.,* feature stacking, without typed functions for explicitly modeling such process; **ii)** their operations are confined to the original coordinate space, causing a heavy computational burden and lacking a global understanding of the given scene; and **iii)** they ignore the adverse effect caused by 'domain gap', *e.g.,* the gap between scene content and motion features. These observations prompted us to think about: *how to empower the optical flow model to effectively obtain the capability of holistic motion understanding?*

To answer this question, we propose to conduct *adaptive graph reasoning* within the conventional optical flow framework. Our idea is to decouple the context reasoning from the matching procedure, and explore scene information to effectively assist in motion estimation by *learning to reason* over the adaptive graph. We argue that a powerful optical flow model should have the capability of going beyond regular grids, which learns to understand the real-world motions under the guidance of scene content from a more global view. Towards this goal, we introduce a novel graph-based approach, namely adaptive graph reasoning for optical flow (AGFlow), which embeds graph techniques onto the matching pipeline to enable an effective context reasoning and information interaction. The proposed AGFlow learns to match features conditioned on scene context, and allows objects' spatial extents to be well aggregated and thus largely decreases the uncertainty of ambiguous matching.

In particular, AGFlow consists of three major components: a motion encoder that maps input frames into high-level representations for computing their 4D correlation volumes, a context encoder that extracts features only from the first input frame for capturing scene information, and an adaptive graph reasoning (AGR) module that learns to understand the given scene and distills useful information to assist in optical flow estimation. Unlike existing approaches (**??**), our AGR exploits scene information for optical flow in the graph domain; that is, both context and motion features are mapped from regular grids to graph space for learning optical flow together. It enables the model to understand the motion from a larger context with only limited extra parameters and FLOPs (see Tab. 3). Importantly, to overcome the domain gap between context and motion representations, a novel graph adapter (GA) is introduced to adapt (motion) graph parameters in a one-shot manner, incorporating the scene information flexibly. Overall, our approach helps optical flow models conduct more efficient scene understanding and naturally distill scene information to assist in optical flow estimation, eventually leading to strong holistic-motion-understanding capability and better performance. The **contributions** of this work are summarized as follows:

- **A novel graph-based approach for optical flow.** To our knowledge, this is the first work that explicitly exploits scene information to assist in optical flow estimation by using graph techniques. The proposed AGFlow can go beyond the regular grids and reason over the graph space to achieve a better motion understanding, thus successfully handling different challenges in optical flow.
- **An adaptive cross-domain graph reasoning approach.** In order to incorporate scene information, we generalize the *learning to adapt* mechanism (**?**) from regular grids to the graph domain. Our designed graph adapter can fast adapt scene context to guide the global (motion) graph reasoning in a one-shot manner.
- **State-of-the-art results on widely-used benchmarks.** Our AGFlow sets new records on both Sintel and KITTI benchmarks, outperforming state-of-the-art approaches by a relatively large margin.

Related Work

Optical Flow Estimation. Optical flow is the task of estimating per-pixel motion between video frames. In the early stage, researchers (**????**) consider this task as an energy minimization problem, with the goal of achieving an ideal tradeoff between feature similarities and motion smoothness. However, as motion itself is hard to be modeled/described by handcrafted features and optimization objectives, it is challenging to obtain precise flow fields by traditional methods. In the deep learning era, early attempts mainly focus on **i)** learning more robust data terms (**??**) or **ii)** avoiding the optimization step to directly estimate optical flow (**??**). To improve results on optical flow, many recent works introduce stronger learning paradigms that can enable iterative refinement (**????**), explicit pixel-wise-relation modeling, and joint representation learning with other tasks (**?**). Although remarkable progress have been achieved by these developments, there is still a large room for improvement over existing approaches that largely focus on addressing the matching similarity between features without considering how to achieve a holistic motion understanding. Taking a further step, our AGFlow is empowered to exploit and incorporate high-level scene information to predict optical flow, linking the low-level matches with high-level semantic information for better accuracy.

Graph Neural Networks. Graph Neural Networks have been widely applied to different applications, including person re-identification (**?**), salient object detection (**?**), 3D shape analysis (**?**), semantic segmentation (**?**) and video question answering (**?**). In the context of optical flow, Graph Convolutional Networks (GCNs) are still largely under-explored. This is because most existing approaches largely focus on the matching similarity between features; the high-level scene information is overlooked or poorly investigated. In this paper, we show that GCNs can benefit the optical flow by comprehensively mining the scene information to assist in flow estimation. It helps the optical model to go beyond

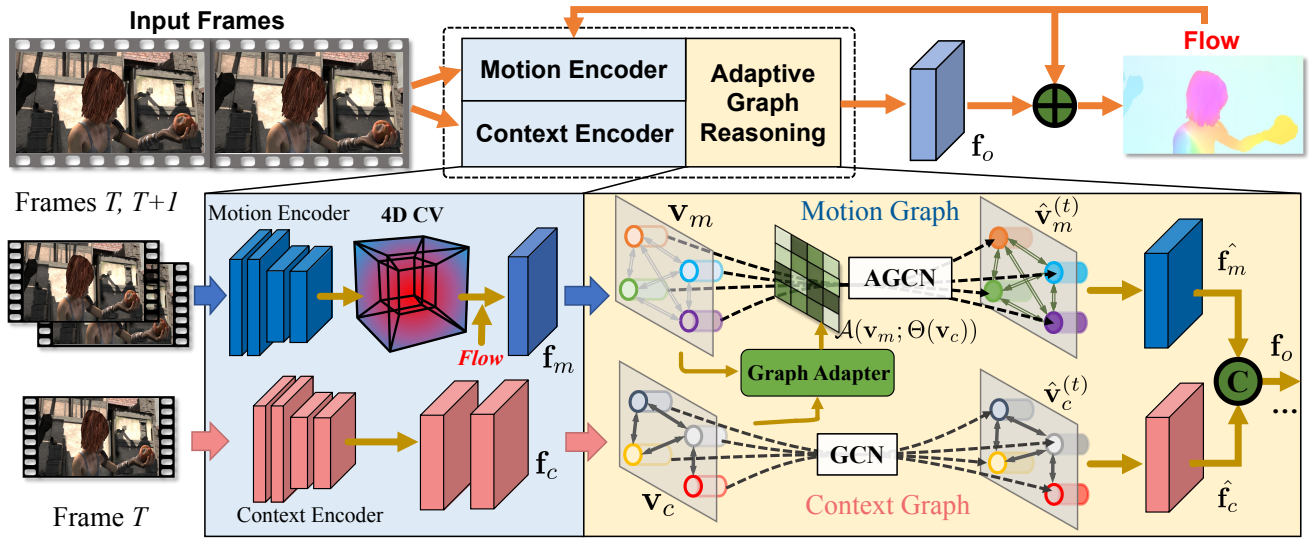

Figure 2: Architecture of the proposed adaptive graph reasoning for optical flow (AGFlow). "4D CV" means 4D correlation volumes, "C" indicates concatenation and "⊕" denotes summation. Please refer to Sec. 3 for more details about the feature notations in this figure. Best viewed in color.

regular grids and understand the motion from a global view. Unlike most conventional GCN-based models (**????**) that only focus on a single domain, our GCN-based AGFlow has the capability of *learning to adapt* cross-domain information, fully incorporating scene information for comprehensive motion understanding.

Methodology

Problem Formulation

Given a pair of input consecutive images, *i.e.*, source image $\mathbf{I}_1$ and target image $\mathbf{I}_2$, the task of optical flow estimation is to predict a dense displacement field between them. Deep learning based flow networks commonly employ an encoder-decoder pipeline to first extract context feature $\mathbf{f}_c$ and obtain motion cues $\mathbf{f}_m$, and then make flow prediction based on the fused feature $\mathbf{f}_o$ in a recurrent/coarse-to-fine manner.

In our approach, we represent the feature fusion in decoder as a graph-based reasoning and learning model, which is formulated as $\mathbf{f}_o = \mathcal{F}_\mathcal{G}(\mathbf{f}_c, \mathbf{f}_m)$. Specifically, we define the model as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ indicates a set of nodes, and $\mathcal{E}$ denotes edges specifying the connection and relation information among nodes. After t runs of graph reasoning, the updated nodes are then mapped back to the original coordinate space to predict the displacement field.

Adaptive Graph Reasoning for Optical Flow

Fig. 2 depicts an overview of the proposed adaptive graph reasoning for optical flow (AGFlow). Following the success of prior work (**?**), we develop our AGFlow based on RAFT (**?**). Specifically, given a pair of input images $\mathbf{I}_1$ and $\mathbf{I}_2$, we employ two residual block based encoders (**?**) to extract a feature pair ($\mathbf{f}_1$, $\mathbf{f}_2$) and context feature $\mathbf{f}_c$. Then, 4D correlation volumes are constructed on the feature pair in four scales. In the recurrent refinement framework, we utilize four convolutions to capture motion feature f_m from the multi-scale matching costs in each 9×9 region. After that, our adaptive graph reasoning (AGR) module takes the motion feature f_m and context feature $\mathbf{f}_c$ as inputs to perform a holistic motion reasoning. Please refer to RAFT (**?**) for more details about the implementations of the baseline model.

Node embedding. The first step is to project context and motion features in regular coordinate space into graph space. The projection operation decouples positional information from the original grid feature, and makes the produced low dimensional node feature representations more compact and with sufficient expressive power. Here we divide the mapped nodes $\mathcal{V}$ in graph model into two groups: *context nodes* $\mathbf{v}_c = \{{v_c}^1, \cdots, {v_c}^\mathrm{n}\}$, containing appearance feature about shape and region information of scene context, and *motion nodes* $\mathbf{v}_m = \{{v_m}^1, \cdots, {v_m}^\mathrm{n}\}$, storing motion feature of cross-image matching dependency.

Specifically, given the context feature $\mathbf{f}_c \in \mathbb{R}^{c \times h \times w}$ and motion feature $\mathbf{f}_m \in \mathbb{R}^{c \times h \times w}$ from encoder network, we employ project function $\mathcal{P}_{\mathrm{f} \to \mathrm{v}}(\cdot)$ to assign features with similar representation to the same node. Let $\mathbf{v}_c \in \mathbb{R}^{C \times K}$ and $\mathbf{v}_m \in \mathbb{R}^{C \times K}$ denote the initial node embeddings in graph space, where C indicates channel number and K is the number of nodes.

To build a global graph over a set of regions, we formulate $\mathcal{P}_{\mathrm{f} \to \mathrm{v}}(\cdot)$ as a linear combination of feature vector in grid space, *i.e.*, $\mathbf{v} = \mathcal{P}_{\mathrm{f} \to \mathrm{v}}(\mathbf{f})$, and thus the produced nodes are able to aggregate long-range information within the overall original feature map. This is given by

$$\mathrm{v}_i = \mathcal{N}(\sum_{\forall j} \mathcal{F}_{\mathrm{f}\to\mathrm{v}}(\mathrm{f})_{ij} \cdot \mathrm{f}_j), \tag{1}$$

where $\mathcal{N}(\cdot)$ is a L-2 normalization function conducted on channel dimension of each node vector, and $\mathcal{F}_{\mathrm{f}\to\mathrm{v}}(\mathrm{f}) \in \mathbb{R}^{N\times K}$ models the projection weights for mapping feature maps to node vectors. Note that our approach can be trained with arbitrary input resolutions. In practice, we first use two convolutions on $\mathbf{f} \in \mathbb{R}^{c\times h\times w}$ to change the channel dimension from c to K so that a feature map with resolution of $K \times h \times w$ can be obtained. Then a reshape function is applied to produce $\mathcal{F}_{\mathrm{f}\to\mathrm{v}}(\mathrm{f})$ with resolution $N \times K$, where $N = h \times w$, and K is a hyperparameter not relying on spatial resolution. Thus the two types of node embedding can be produced by $\mathbf{v}_c = \mathcal{P}_{\mathrm{f}\to\mathrm{v}}(\mathbf{f}_c)$, and $\mathbf{v}_m = \mathcal{P}_{\mathrm{f}\to\mathrm{v}}(\mathbf{f}_m)$.

Adaptive Graph Reasoning. Given node embeddings $\mathbf{v}$ in graph space, the adjacency matrix for graph reasoning can be commonly generated by measuring the similarity among all node vectors (**?**), as $\mathbf{A} = \mathbf{v}^{\mathsf{T}}\mathbf{v}$. After modeling the adjacency matrix $\mathbf{A}$, the graph reasoning with graph convolutional network (**?**) is defined as

$$\hat{\mathbf{v}} = \mathcal{F}_{\mathtt{G}}(\mathbf{v}, \mathbf{A}) = \sigma(\mathbf{A}\mathbf{v}^{\mathsf{T}}\mathrm{w}_{\mathtt{G}}), \tag{2}$$

where $\sigma(\cdot)$ is a non-linear activation function, and $\mathrm{w}_{\mathtt{G}}$ is learnable parameters for graph convolution. $\hat{\mathbf{v}}$ is the updated node representations with graph reasoning, and it can be iteratively enhanced with more runs as $\hat{\mathbf{v}}^{(t)} = \mathcal{F}_{\mathtt{G}}(\mathbf{v}, \mathbf{A})^{(t)}$, where t denotes update iterations.

Let us consider the representation property of context and motion nodes. Motion nodes mainly encode the pointwise correspondence between an image pair while neglecting the intra-relations among pixels within regions, the context nodes, on the contrary, capture discriminative features for region and shape representations. Thus, we need to address two hurdles: First, there is an inevitable representation gap between context and motion nodes, which could hinder the effective information propagating of directly overall graph reasoning. Second, motion nodes lack constraints on shape or layout for the potential displacement field, and thus they are unable to yield enough context information for individual graph reasoning.

To tackle the issues, we propose an *adaptive graph reasoning* (AGR) module to decouple the context reasoning from the matching procedure, and simultaneously transfer the region and shape prior of scene context to motion nodes in a one-shot manner. The key idea is to exploit the discriminative representations of shape and region in global context to guide the learning of motion adjacency matrix with adaptive parameters. Inspired by (**?**), we devise an adaptive procedure of adjacency matrix learning to predict dynamic parameters that tailor the motion relation modeling based on image-specific contextual information. This is given by

$$\breve{\mathbf{A}} = \mathcal{A}(\mathbf{v}_m; \Theta(\mathbf{v}_c)), \tag{3}$$

where $\Theta(\cdot)$ is a parameter learner, and $\mathcal{A}(\cdot)$ denotes a context-to-motion graph adapter (GA) equipped with dynamic weights from $\Theta(\mathbf{v}_c)$. In practice, we implement the

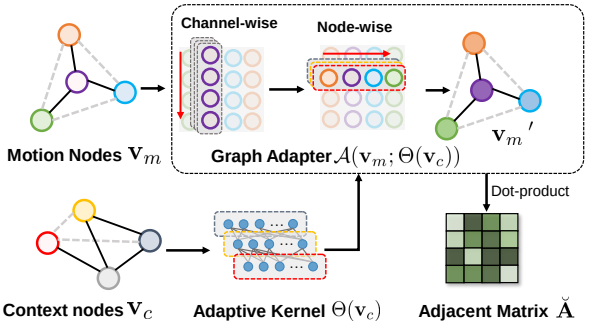

Figure 3: Architecture of Graph Adapter.

$\Theta(\cdot)$ with a linear projection function with a softmax activation. As shown in Fig. 3, $\mathcal{A}(\cdot)$ is implemented with a two-layer MLP, where the first regular linear function followed by a ReLU activation is applied to perform channel-wise learning, and then the second linear function with adaptive kernel $\Theta(\mathbf{v}_c)$ is utilized to perform node-wise interaction for relation learning with context-to-motion adaptation. Specifically, given context nodes $\mathbf{v}_c \in \mathbb{R}^{C\times K}$, we predict the adaptive kernel $\Theta(\mathbf{v}_c) \in \mathbb{R}^{K\times K}$ with convolutions on channel dimension ($C \to K$). Then, we transfer it into the adaptive weights with shape $K \times K$ of the second linear function, which is used to produce $\mathbf{v}_m{}'$. Finally, we apply the dot-product similarity on $\mathbf{v}_m{}'$ to predict $\breve{\mathbf{A}}$.

The produced parameter $\Theta(\mathbf{v}_c)$ relies on context nodes for dynamically leveraging the shape and region information of current input. Thus, motion nodes can be fast adaptive to scene context and favorably make full use of the transferred node relation for motion sub-graph reasoning. Thus the enhanced context nodes $\hat{\mathbf{v}}_c^{(t)}$ are produced by

$$\hat{\mathbf{v}}_c^{(t)} = \mathcal{F}_{\mathtt{G}}(\mathbf{v}_c, \mathbf{A})^{(t)}, \mathtt{where}\ \mathbf{A} = \mathbf{v}_c^{\mathsf{T}}\mathbf{v}_c, \tag{4}$$

and similarly, motion nodes $\hat{\mathbf{v}}_m^{(t)}$ are produced by

$$\hat{\mathbf{v}}_m^{(t)} = \mathcal{F}_{\mathtt{AG}}(\mathbf{v}_m, \breve{\mathbf{A}})^{(t)}, \mathtt{where}\ \breve{\mathbf{A}} = \mathcal{A}(\mathbf{v}_m; \Theta(\mathbf{v}_c)), \tag{5}$$

and $\mathcal{F}_{\mathtt{AG}}(\cdot)$ denotes motion nodes reasoning with adaptive graph convolutional network (AGCN).

Attentive Readout. After t runs of relation reasoning and state updating, we present an attentive readout module to project the enhanced context nodes $\mathbf{v}_c^{(t)}$ and motion nodes $\hat{\mathbf{v}}_m^{(t)}$ from graph space back to grid feature space, making the overall graph interaction model compatible with existing flow networks. Then the updated feature maps contain both global contextual information and local pixel-wise matching cost, which are properly used to make a better prediction for the flow field.

In particular, we formulate the reverse projection as

$$\hat{\mathbf{f}} = \mathcal{P}_{\mathrm{v}\to\mathrm{f}}(\hat{\mathbf{v}}), \tag{6}$$

where $\mathcal{P}_{\mathrm{v}\to\mathrm{f}}(\cdot)$ is a linear combination function that map the node vectors $\hat{\mathbf{v}} \in \mathbb{R}^{C\times K}$ to the feature maps $\hat{\mathbf{f}} \in \mathbb{R}^{C\times N}$

Training Data	Method	Sintel (val)		KITTI-15 (val)		Sintel (test)		KITTI-15 (test)
		Clean	Final	EPE	F1-all	Clean	Final	F1-all
C + T	HD3(**?**)	3.84	8.77	13.17	24.0	-	-	-
	LiteFlowNet(**?**)	2.48	4.04	10.39	28.5	-	-	-
	PWC-Net(**?**)	2.55	3.93	10.35	33.7	-	-	-
	FlowNet2(**?**)	2.02	3.54	10.08	30.0	3.96	6.02	-
	DICL(**?**)	1.94	3.77	8.70	23.6	-	-	-
	RAFT(**?**)	1.43	2.71	5.04	17.4	-	-	-
	SCV(**?**)	**1.29**	2.95	6.80	19.3	-	-	-
	AGFlow (ours)	1.31	**2.69**	**4.82**	**17.0**	-	-	-
C + T + S + K (+ H)	PWC-Net+(**?**)	(1.71)	(2.34)	(1.50)	(5.3)	3.45	4.60	7.72
	IRR-PWC (**?**)	(1.92)	(2.51)	(1.63)	(5.3)	3.84	4.58	7.65
	VCN (**?**)	(1.66)	(2.24)	(1.16)	(4.1)	2.81	4.40	6.30
	MaskFlowNet(**?**)	-	-	-	-	2.52	4.17	6.10
	ScopeFlow(**?**)	-	-	-	-	3.59	4.10	6.82
	DICL(**?**)	(1.11)	(1.60)	(1.02)	(3.6)	2.12	3.44	6.31
	RAFT(**?**)	(0.76)	(1.22)	(0.63)	(1.5)	1.61*	2.86*	5.10
	SCV(**?**)	(0.79)	(1.70)	(0.75)	(2.1)	1.77*	3.88*	6.17
	AGFlow (ours)	**(0.65)**	**(1.07)**	**(0.58)**	**(1.2)**	**1.43***	**2.47***	**4.89**

Table 1: Quantitative comparison with state-of-the-art methods using EPE and F1-all metrics (the lower the better). Following previous works (**???**), we compare our results with all published works on three passes from two standard benchmarks. "C + T" indicates models are pretrained on FlyingChairs(C) and FlyingThing(T) to test generalization performance. "+ S + K (+ H)" denotes the training data combining Sintel(S), KITTI(K) and HD1K(H). "+H" with brackets means it is optional for some works (**??**). "*" denotes the results with warm-start testing (**?**). The best results are marked in **bold** for better comparison.

in the original grid space of flow network. In practice, we reuse the projection matrix in the node embedding procedure. The projection matrix contains pixel-to-node assignments and preserves the spatial details, which is crucial for recovering the resolution of feature maps. Besides, no additional parameters are involved by reusing the region assignments, which also helps to reduce computational overhead.

The context feature $\hat{\mathbf{f}}_c$ is produced by a residual operation as

$$\hat{\mathbf{f}}_c = \mathbf{f}_c + \alpha \mathcal{P}_{\mathrm{v}\to\mathrm{f}}(\mathbf{v}_c), \tag{7}$$

where α denotes a learnable parameter that is initialized as 0 and gradually performs a weighted sum. Similarly, motion feature $\hat{\mathbf{f}}_m$ is produced by

$$\hat{\mathbf{f}}_m = \mathbf{f}_m + \beta \mathcal{P}_{\mathrm{v}\to\mathrm{f}}(\mathbf{v}_m). \tag{8}$$

Given the enhanced feature $\hat{\mathbf{f}}_c$ and $\hat{\mathbf{f}}_m$, one potential hurdle for feature fusion is that context feature lacks correspondence information for cross-image matching, which could lead to a shift of global displacement and thus affect the flow accuracy. Therefore, we devise an attentive fusion function, which first learns to predict a set of scale weights from motion feature $\hat{\mathbf{f}}_m$ and then leverages them to implement global adjustment for entire dense displacement. Specifically, the attentive fusion function is defined as

$$\mathbf{f}_o = (1 + \mathcal{F}_{\mathtt{CA}}(\hat{\mathbf{f}}_m))\hat{\mathbf{f}}_c \oplus \hat{\mathbf{f}}_m, \tag{9}$$

where $\oplus$ is a concatenation operation, and $\mathcal{F}_{\mathtt{CA}}(\cdot)$ is a channel attention function (**?**), here implemented with two convolutions with ReLU and sigmoid activation.

Experimental Results

Datasets and Evaluation Metrics

We conduct extensive experiments on two standard datasets, *i.e.,* MPI-Sintel (**?**) and KITTI 2015 (**?**). We follow prior works (**??**) to utilize two standard evaluation metrics, *i.e.*, average end-point error (EPE) and the percentage of erroneous pixels > 3 pixels (F1-all), to evaluate the performance of predicted optical flow.

Implementation Details

The implementation of our approach is based on PyTorch toolbox. In our model, we set the number of context and motion nodes K to 128. The state updating iterations t are set to 2 and 1 for context and motion graph, respectively.

During training, we follow prior works (**??**) to adopt AdamW optimizer with one-cycle learning rate policy, and conduct model pretraining on synthetic data as the standard optical flow training procedure. The model is pretrained on FlyingChairs (**?**) for 180k iterations and then on FlyingThings (**?**) for 180k iterations. After that, we fine-tune the model on combined data from Sintel (**?**), KITTI-2015 (**?**), and HD1K (**?**) for 180k iterations, and then submit the flow prediction to Sintel server for online evaluation. Finally, additional 50k iterations of finetuning are performed on KITTI-2015 (**?**) for KITTI online evaluation. Our model is trained on 2 NVIDIA GeForce GTX 2080Ti GPUs, and the batch size is set to 8 for better leveraging the GPU memory.

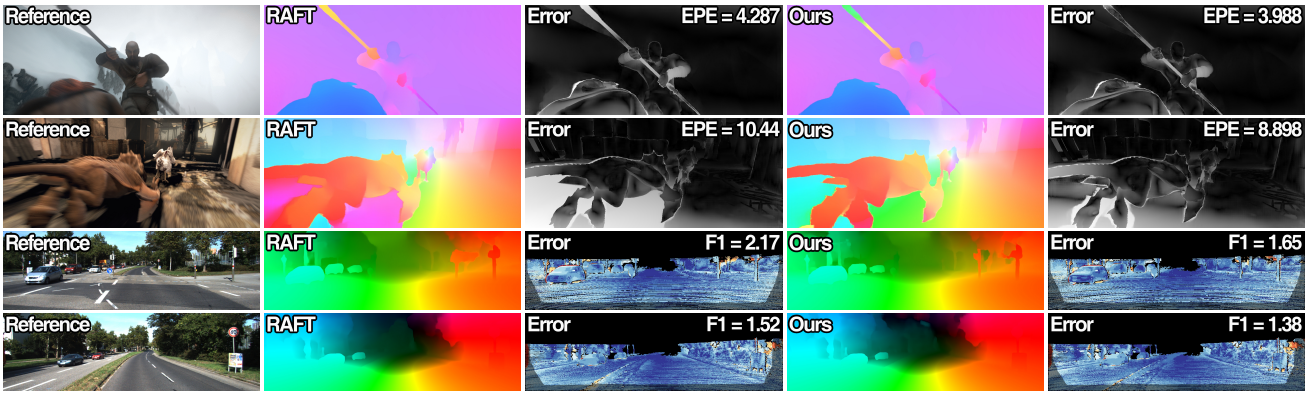

Figure 4: Qualitative comparisons with RAFT (**?**) on Sintel and KITTI `test` set. All results are provided by the official website of each dataset. Best viewed in color.

Comparison with State-of-the-Arts

Results on Sintel. On the training set of FlyingChairs (C) + FlyingThings (T), as shown in Tab. 1, our approach achieves an average EPE of 1.31 on clean pass of Sintel dataset, which is competitive with SCV (**?**) and lower than the well-known RAFT (**?**) by 8.4% (1.43 → 1.31). On final pass, it obtains a score of 2.69 in EPE, outperforming previous state-of-the-art methods SCV and RAFT by 8.8% (2.95 → 2.69) and 0.7% (2.71 → 2.69), respectively. The results demonstrate the good cross dataset generalization of our model.

On Sintel test set, we follow prior works (**???**) to submit the predicted flow to the official server for online evaluation. Our AGFlow achieves an EPE of 1.68 on on Sintel clean pass, which surpasses top-ranked methods RAFT (**?**) and SCV (**?**) by 13.4% (1.94 → 1.68) and 2.3% (1.72 → 1.68), respectively. Besides, it obtains EPE = 2.83 on final pass, outperforming recent SCV by a large margin (21.4%). When utilizing the warm-start strategy, our approach sets new records of 1.43 EPE on clean pass and 2.47 on final pass, which significantly surpasses the previous best results by 11.2% (1.61 → 1.43) and 13.6% (2.86 → 2.47), respectively. It is worth noting that our AGFlow ranks 1st on final pass of Sintel benchmark among all approaches at the time of submission.

Fig. 4 (line 1 and 2) provides some qualitative comparisons with RAFT (**?**) on the challenging final pass of Sintel dataset, which demonstrates that our AGFlow is able to fully exploit scene context to effectively assist motion estimation with shape and region constraints, leading to achieving more accurate flow fields with clear motion boundaries.

Results on KITTI. We also provide the results of our approach on KITTI-15 dataset. As in Tab. 1, when training on C + T, our AGFlow achieves an average EPE of 4.82 and F1-all score of 17.0% on KITTI-15 validation set, which significantly surpass recent method SCV (**?**) by 29.1% (6.80 → 4.82) and 11.9% (19.3 → 17.0), respectively. For online evaluation, our approach achieve new state-of-the-art performance of 4.89% in F1-all, outperforming top-ranked methods SCV (**?**) and RAFT (**?**) by 20.7% (6.17 → 4.89) and 4.1% (5.10 → 4.89), respectively. Some qualitative comparisons with RAFT (**?**) on KITTI dataset are illustrated in Fig. 4 (line 3 and 4), which shows that the proposed global reasoning conditioned on scene context helps to decrease the uncertainty of ambiguous matching in some tough regions.

Ablation Analysis

Comparison with Grid Feature Enhancement. We first compare the proposed AGFlow with widely-used methods for optical flow that enhance features in regular grid space, including RAFT (**?**), dense and dilated convolutions (**??**). As shown in Tab. 2, dense and dilated convolutions slightly improve the flow accuracy with heavy model complexity. In contrast, our AGFlow achieves better performance, yet only needs additional 0.30 M parameters, reducing parameters by around 90%. This demonstrates that the proposed low dimensional graph reasoning scheme is effective to boost the flow accuracy in an efficient manner.

Comparison with SuperGlue (?). SuperGlue is a well-known method that employs graph neural network for feature matching. As can be seen in Tab. 2 (line 4), the original SuperGlue on grid feature requires a large amount of GPU memory, which is out of range with general settings for model training. Thus we re-implement SuperGlue with our low dimensional graph model (termed G-SuperGlue). Compared with G-SuperGlue, our AGFlow not only achieves considerable performance gain (8.0%), but also reduces parameters by 85%. This is because our adaptive graph reasoning scheme is simple yet effective, and capable of fast transferring the region and shape prior from scene context to motion nodes in a one-shot manner.

Comparison with GCU (?). We also compare our AGFlow with the basic region-based graph reasoning model (**?**). Since GCU projects all feature maps into a single type of node, it requires fewer parameters than other methods. However, as we mention above, the representation gap between context and motion feature hinders the effectiveness of relation reasoning with a simple graph, thus resulting

	Method	Param (M) ↓	Sintel (val) EPE ↓	
			Clean	Final
Grid Feature	RAFT	5.26	1.65	3.04
	+ Dense Convs	+ 3.34	1.63	2.98
	+ Dilated Convs	+ 0.85	1.66	3.02
	+ SuperGlue	+ 2.99	-	-
Graph Space	GCU (Base)	5.44	1.76	3.14
	+ G-SuperGlue	+ 1.82	1.63	3.01
	+ **AGR (ours)**	+ 0.12	**1.50**	**2.88**

Table 2: Quantitative comparisons with related methods (refer to Sec. for more details). We set RAFT (**?**) as the baseline model for the method with regular gird space enhancement. The methods in each part are plugged into the same baseline and trained on C + T ($180k$) for fair comparison.

	Settings	FLOPs (G)	Param (M)	Sintel (val) EPE	
				Clean	Final
Graph Reasoning	Base Graph	381.06	5.44	1.76	3.14
	+ SGR (no GA)	+ 16.95	+ 0.07	1.65	2.97
	+ AGR	+ 17.09	+ 0.12	1.50	2.88
Node numbers	$K = 32$	+ 3.09	+ 0.04	1.61	3.02
	$K = 64$	+ 5.88	+ 0.07	1.56	2.95
	$K = 128$	+ 17.09	+ 0.12	1.50	2.88
	$K = 256$	+ 61.84	+ 0.2	1.49	2.90
Attentive readout	On	+ 17.09	+ 0.12	1.50	2.88
	Off	+ 16.82	+ 0.10	1.55	2.94

Table 3: Ablation analysis for different settings of our AGFlow. "SGR" indicates separated graph reasoning for context and motion nodes (*i.e.*, without graph adapter), and "AGR" denotes overall adaptive graph reasoning. All methods are trained on C + T ($180k$) for fair comparison. Underline indicates the default settings in our model.

in a performance drop on both passes of Sintel. We regard it as the base graph model, as in line 5 of Tab. 2. In contrast, we carefully project feature maps into context and motion nodes, and further propose an adaptive graph reasoning approach to perform the *task-specific* hybrid reasoning, allowing our model to significantly reduce the average end-point error by around 14.8%.

Effectiveness of Adaptive Graph Reasoning. In Tab. 3, we empirically analyze the computational cost and corresponding performance gain of the core component in the proposed AGFlow. Using separated context and motion reasoning based on graph model boosts the performance by $5.4\% \sim 6.3\%$. Besides, we further incorporate the proposed graph adapter into the context-to-motion interaction, and then yield the proposed adaptive graph reasoning (AGR) module, which brings about 9% additional performance gain with only 0.14 G FLOPs and 0.05 M parameters for extra computational overhead.

Ablation for Node Numbers. We empirically show the influences of node numbers K in our graph model. As shown in Tab. 3, when more nodes are used ($32 \rightarrow 64 \rightarrow 128$), the average end-point error are gradually decreased from 1.61 on Sintel clean pass and 3.02 on final pass to 1.50 and 2.88, respectively. However, if furthermore nodes are involved ($128 \rightarrow 256$), the flow accuracy almost remains the same and the computational overhead is largely increased by 2.6 times. This is because some redundant feature representations are generated with nodes, which brings no benefit to flow estimation. Therefore, we set $K = 128$ to ensure a good balance between efficiency and performance.

Method	Param (M) ↓	Time (ms) ↓
RAFT	5.26 (-)	86.9 (-)
GMA	5.8 (+ 0.54)	113.8 (+ 26.9)
AGFlow	5.56 (+ 0.30)	90.7 (+ 3.8)

Table 4: Computational comparisons with state-of-the-arts on a single Geforce RTX 2080Ti GPU.

Ablation for Attentive Readout. We also test the influence of attentive readout compared with regular readouts (**?**) in Tab. 3. As can be seen, incorporating it into our model brings about 3% in performance gain and only requires negligible computation cost and parameters, demonstrating the cost-effective property of this component.

Runtime Comparison. We provide the parameters and runtime of state-of-the-art methods in Tab. 4. Compared with GMA (**?**), our AGFlow can achieve competitive performance while reducing 0.24 M parameters. Besides, the inference speed is boosted by 20.3% ($113.8 \rightarrow 90.7$). The comparisons clearly demonstrate the effectiveness of our AGFlow.

Conclusion

In this paper, we present a novel graph-based approach termed adaptive graph reasoning for optical flow (AGFlow), which performs global reasoning to explicitly emphasize scene context and motion dependencies for flow estimation. The key idea is adaptive graph reasoning, which intends to fast enhance the feature representation of motion nodes conditioned on the global context with shape and boundary. Comprehensive experiments demonstrate that our AGFlow is effective and flexible to alleviate the matching ambiguities in challenging scenes, and sets new records in two standard flow benchmarks. We hope our work will offer a fresh perspective in re-thinking the design of optical flow models.

Acknowledgements. This work was supported by the National Key R&D Plan of the Ministry of Science and Technology No. 2020AAA0104400, and the National Natural Science Foundation of China (NSFC) No.61872067 and No.61720106004.

Ab repellendus molestiae odio odit libero minima ipsa numquam ipsam voluptatem pariatur, delectus repellendus labore voluptas perferendis quia id minus veritatis eligendi corrupti modi, est accusamus ipsa quisquam minus, enim et quod dignissimos cupiditate eveniet pariatur optio earum voluptatem aspernatur corrupti, consequuntur magnam nihil harum?Recusandae consequatur dolor accusantium libero nisi necessitatibus natus, libero tempora ratione illum obcaecati optio quis beatae, dolore officia maiores similique, nobis quod culpa, nobis voluptatem tempore quam?Error sequi illo odit officia explicabo dicta, tenetur omnis sunt eum nihil mollitia fugit magni?