



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 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 meth-

ods 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 |
| | + Dense Convs | + 3.34 | 1.63 |
| | + Dilated Convs + SuperGlue | + 0.85 + 2.99 | 1.66 - |
| Graph Space | GCU (Base) | 5.44 | 1.76 |
| | + G-SuperGlue | + 1.82 | 1.63 |
| | + 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 grid 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 |
| | <u>+ AGR</u> | + 17.09 | + 0.12 | 1.50 | 2.88 |
| Node numbers | <u>K</u> = 32 | + 3.09 | + 0.04 | 1.61 | 3.02 |
| | <u>K</u> = 64 | + 5.88 | + 0.07 | 1.56 | 2.95 |
| | <u>K</u> = 128 | + 17.09 | + 0.12 | 1.50 | 2.88 |
| | <u>K</u> = 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% ~ 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 → 64 → 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 → 256), the flow accuracy almost remains the

| 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.

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.

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 → 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?