What Makes RAFT Better Than PWC-Net?

Deqing Sun∗†, Michael Rubinstein
Charles Herrmann∗ David Fleet
Fitsum Reda William T. Freeman
Google Research

Abstract

How important are training details and datasets to recent optical flow models like RAFT? And do they generalize? To explore these questions, rather than develop a new model, we revisit three prominent models, PWC-Net, IRR-PWC and RAFT, with a common set of modern training techniques and datasets, and observe significant performance gains, demonstrating the importance and generality of these training details. Our newly trained PWC-Net and IRR-PWC models show surprisingly large improvements, up to 30% versus original published results on Sintel and KITTI 2015 benchmarks. They outperform the more recent Flow1D on KITTI 2015 while being 3× faster during inference. Our newly trained RAFT achieves an Fl-all score of 4.31% on KITTI 2015, more accurate than all published optical flow methods at the time of writing. Our results demonstrate the benefits of separating the contributions of models, training techniques and datasets when analyzing performance gains of optical flow methods. Our source code will be publicly available.

1 Introduction

The field of optical flow has witnessed rapid progress in recent years, driven largely by deep learning. FlowNet [9] first demonstrated the potential of deep learning for optical flow, while PWC-Net [42] was the first model to eclipse classical flow techniques. The widely-acclaimed RAFT model [45] reduced error rates on common benchmarks by up to 30% versus state-of-the-art baselines, outperforming PWC-Net by a wide margin. RAFT quickly became the predominant framework for optical flow [19, 26, 29, 36, 48, 51, 52, 58] and related tasks [23, 46].

The success of RAFT has been attributed primarily to its novel network design, including its multi-scale all-pairs cost volume, its recurrent update operator, and its up-sampling module. At the same time, other factors like training procedures and datasets have also evolved, and may play important roles. In this work, we pose the question: How much do training techniques of recent methods like RAFT contribute to their impressive performance? And, importantly, can these training innovations similarly improve the performance of other models?

We begin by revisiting the 2018 PWC-Net [42], and investigate the impact of datasets and training techniques for both pre-training and fine-tuning. We show that, even with such a relatively “old” model, by employing recent datasets and advances in training, and without any changes to the originally proposed architecture, one can obtain substantial performance gains, outperforming more recent models [52, 60] and resolving finer-grained details of flow fields (see, e.g., Fig. 1 and Table 1).

We further show that the same enhancements yield similar performance gains when applied to IRR-PWC, a prominent variant of PWC-Net that is closely related to RAFT. Indeed, these insights also yield an improved version of RAFT, outperforming all published optical flow models on the KITTI 2015 benchmark at the time of writing.

We make the following contributions:

∗ Equal technical contribution, † project lead
Figure 1: Left: Large improvements with newly trained PWC-Net, IRR-PWC and RAFT (left: originally published results in blue; results of our newly trained models in red). The newly trained RAFT is more accurate than all published methods on KITTI 2015 at the time of writing. Right: Visual comparison on a Davis sequence between the original [43] and our newly trained PWC-Net and RAFT, shows improved flow details, e.g. the hole between the cart and the person at the back. The newly trained PWC-Net recovers the hole between the cart and the front person better than RAFT.

- We show that a newly trained PWC-Net, using ingredients from recent training techniques (gradient clipping, OneCycle learning rate, and long training) and modern datasets (AutoFlow), yields surprisingly competitive results on Sintel and KITTI benchmarks.
- These same techniques also deliver sizeable performance gains with two other prominent models, IRR-PWC and RAFT. Our newly trained RAFT is more accurate than all published optical flow methods on KITTI 2015.
- We perform a thorough ablation study on pre-training and fine-tuning to understand which ingredients are key to these performance improvements and how they are manifesting.
- The newly trained PWC-Net and IRR-PWC produce visually good results on 4K Davis input images, making them an appealing option for applications that require fast inference with low memory overhead.

To enable fair comparisons and facilitate innovations, our code and pre-trained models will be made public.

2 Previous Work

Deep models for optical flow. FlowNet [9] was the first model to demonstrate the potential of deep learning for optical flow, and inspired various new architectures. FlowNet2 [17] stacked basic models to improve model capacity and performance, while SpyNet [32] used an image pyramid and warping to build a compact model. PWC-Net [42] used classical optical flow principles (e.g., [3, 40, 44]) to build an effective model, which has since seen widespread use [2, 7, 18, 21, 33, 39, 59, 61]. The concurrent LiteFlowNet [15] used similar ideas to build a lightweight network. TVNet [10] took a different approach with classical flow principles by unrolling the optimization iterations of the TV-L1 method [57].

Many architectures have used a pyramid structure. IRR-PWC [16] introduced iterative refinement, reusing the same flow decoder module at different pyramidal levels. VCN [53] used a 4D cost volume that is easily adapted to stereo and optical flow. HD3 [55] modeled flow uncertainty hierarchically. MaskFlowNet [60] jointly modeled occlusion and optical flow. Improvements brought by each model over the previous SOTA was often within 5% on Sintel (c.f., Table 1).

A recent, notable architecture, RAFT [45], built a full cost volume and performs recurrent refinements at a single resolution. RAFT achieved a significant improvement over previous models on Sintel and KITTI benchmarks, and became a starting point for numerous new variants [19, 26, 29, 36, 48, 51, 52, 58]. To reduce the memory cost of the all-pairs cost volume, Flow1D used 1D self-attention with 1D search, with minimal performance drop while enabling application to 4K video inputs [52]. SeparableFlow used a non-local aggregation module for cost aggregation, yielding substantial performance gains [58].

2
Table 1: Results of 2-frame methods on public benchmarks (AEPE↓ for Sintel and Fl-all↓ for KITTI). Bold indicates the best number and underline the second-best. The running time is for $448 \times 1024$ resolution input (*reported in paper); the differences will be larger for higher resolution (c.f. Table 6). Newly trained PWC-Net, IRR-PWC and RAFT are substantially more accurate than their predecessors. With improved training protocols, the newly trained PWC-Net and IRR-PWC are more accurate than some recent methods [52, 53] on KITTI 2015 while being about $3 \times$ faster in inference. The newly trained RAFT is more accurate than all published methods on KITTI 2015.

Datasets for optical flow. For pre-training the predominant dataset is FlyingChairs [9]. Ilg et al. [17] introduced a dataset schedule that uses FlyingChairs and FlyingThings3D [28] sequentially. This remains a standard way to pre-train models. Sun et al. [41] proposed a new dataset, AutoFlow, which learns rendering hyperparameters and shows moderate improvements over the FlyingChairs and FlyingThings3D in pre-training PWC-Net and RAFT. For fine-tuning, the limited training data from Sintel and KITTI are often combined with additional datasets, such as HD1K [22] and VIPER [54], to improve generalization. In this paper, we show that PWC-Net and its variant, IRR-PWC, obtain competitive results when pre-trained on AutoFlow and fine-tuned using recent techniques.

Training techniques for optical flow. While different papers tend to adopt slightly different training techniques and implementation details, some have examined the impact of recent training techniques on older architectures. Ilg et al. [17] found that using dataset scheduling can improve the pre-training results of FlowNetS and FlowNetC. Sun et al. [43] obtained better fine-tuning results with FlowNetS and FlowNetC on Sintel by using improved data augmentation and learning rate disruption; they also improved on the initial PWC-Net [42] by using additional datasets. Sun et al. [41] reported better pre-training results for PWC-Net but did not investigate fine-tuning. Here, with PWC-Net, IRR-PWC and RAFT, we show significantly better fine-tuning results.

Self-supervised learning for optical flow. Significant progress has been achieved with self-supervised learning for optical flow [20, 24, 30, 38, 56], focusing more on the loss than model architecture. UFlow [20] systematically studied a set of key components for self-supervised optical flow, including both model elements and training techniques. Their study used PWC-Net as the
main backbone. Here we focus on training techniques and datasets, systematically studying three prominent models to identify factors that generalize across models.

**Similar study on other vision tasks.** The field of classification has also started to more closely examine whether performance improvements in recent papers come from the model architecture or training details. Both [14] and [50] examined modern training techniques on ResNet-50 [13] and observed significant performance improvements on ImageNet [6], improving top-1 precision from 76.2 in 2015, to 79.3 in 2018, and finally to 80.4 in 2021. These gains have come solely from improved training details, namely, from augmentations, optimizers, learning rate schedules, and regularization. The introduction of vision transformers (ViT) [8] also led to a series of papers [37, 47] on improved training strategies, substantially improving performance from the initial accuracy of 76.5 up to 81.8.

Other recent papers took a related but slightly different direction, simultaneously modernizing both the training details and architectural elements but cleanly ablating and analyzing the improvements. Bello et al. [4] included an improved training procedure as well as exploration of squeeze-and-excite and different layer changes. Liu et al. [25] used recent training details and iteratively improves ResNet with modern network design elements, improving the accuracy from 76.2 to 82.0, which is competitive with similarly sized state-of-the-art models. While these papers mainly studied a single model and often involved modifying the backbone, we investigate three different models to understand key factors that apply to different models, and the trade-offs between models.

### 3 Approach and Results

Our goal is to understand which innovations in training techniques, principally from RAFT, play a major role in the impressive performance of modern optical flow methods, and to what extent they generalize well to different architectures. To this end, we decouple the contributions of model, training techniques, and dataset, and perform comparisons by changing one variable at a time. More specifically, we revisit PWC-Net, IRR-PWC and RAFT with the recently improved training techniques and datasets. Ablations are used to determine which factors are responsible for performance improvements. In doing so, we consider various factors including pre-training, fine-tuning, training duration, memory requirements and inference speed.

#### 3.1 Models Evaluated

The first model we evaluate is PWC-Net, the design of which was inspired by three classical optical flow principles, namely pyramids, warping, and cost volumes. These inductive biases make the network effective, efficient, and compact compared to prior work. IRR-PWC [16] introduces iterative refinement and shares the optical flow estimation network weights among different pyramid levels. The number of iterative refinement steps for IRR-PWC is the number of pyramid levels. RAFT is closely related to IRR but enables an arbitrarily large number of refinement iterations. It has several novel network design elements, such as the recurrent refinement unit and convex upsampling module. Notably, RAFT eschews the pyramidal refinement structure, instead using an all-pairs cost volume at a single resolution.

**Memory usage.** For an $H \times W$ input image, the memory cost for constructing the cost volume in RAFT is $O((HW)^2D)$, where $D$ is the number of feature channels (constant, typically 256 for RAFT and $\leq 192$ for PWC-Net and IRR-PWC). To reduce the memory cost for high-resolution inputs, Flow1D constructs a 1D cost volume with cost of $O(HW(H+W)D)$. By comparison, the memory needed for the cost volume in PWC-Net and IRR-PWC is $O(HWD(2d+1)^2)$, where the constant $d$ is the search radius at each pyramid level (default 4). Note that $(2d+1)^2 \ll H+W \ll HW$ for high-resolution inputs; this is particularly important for 4K videos, which are becoming increasingly popular. We empirically compare memory usage at different resolutions in Table 6.

#### 3.2 Pre-training

Optical flow models are often pre-trained on synthetic datasets, such as FlyingChairs, and then fine-tuned using a small amount of training data for the Sintel and KITTI benchmarks. Since the introduction of PWC-Net in 2018, new training techniques and datasets have been proposed. As
shown in [43], better training techniques and new datasets improve the pre-training performance of PWC-Net. We investigate how PWC-Net and IRR-PWC performs with the same pre-training procedure, and whether the procedure can be further improved.

Table 2 summarizes the results of pre-training PWC-Net, IRR-PWC and RAFT using different datasets and techniques. (To save space, we omit some results for PWC-Net and RAFT and refer readers to [41] for details.) We further perform an ablation study on several key design choices using PWC-Net, the results of which are shown in Table 3. To reduce the effects of random initialization, we independently train the model six times, and report the results of the best run. While the original IRR-PWC computes bidirectional optical flow and jointly reasons about occlusion, we test a lightweight implementation without bidirectional flow and occlusion reasoning.

**Pre-training datasets.** Pre-training using AutoFlow results in significantly better results than FlyingChairs for PWC-Net, IRR-PWC and RAFT. Figure 2 visually compares the results by two PWC-Net models on Davis [31], and Middlebury [1] sequences. PWC-Net trained on AutoFlow better recovers fine motion details (top) and produces coherent motion for the foreground objects (bottom).

![Figure 2: Visual results of PWC-Net pre-trained using FlyingChairs and AutoFlow on Davis and Middlebury input images. PWC-Net trained using AutoFlow recovers fine details between the legs (top) and coherent motion for the girl and the dog (bottom).](image)

**Gradient clipping.** Gradient clipping is a heuristic to avoid cliff structures for recurrent neural networks [12]. The update operator of RAFT uses a GRU block that is similar to the LSTM block. Thus, RAFT training uses gradient clipping to avoid exploding gradients. Using gradient clipping during the training process also improves the performance of PWC-Net and IRR-PWC substantially and it results in more stable training. Removing gradient clipping from RAFT results in moderate performance degradation. We perform an ablation study on the threshold of gradient clipping and find that the training is quite robust to this parameter value (Table 3).

**Learning rate schedule.** Before RAFT, nearly all optical flow models have been trained using a piecewise learning rate, with optional learning rate disruption. RAFT uses a OneCycle learning rate schedule, which starts from a small learning rate, linearly increases to the peak learning rate, and then linearly decreases to the starting learning rate. Using the OneCycle learning rate improves the performance of all three models (Table 2). Moving the position of the peak toward the origin slightly improves the performance (Table 3). Note that, for other published models, those that use gradient clipping and the OneCycle learning rate, e.g., Flow1D and SeparableFlow, are generally better than those that do not, e.g., VCN and MaskFlowNet. It would be interesting, though outside the scope of this paper, to investigate the performance of VCN and MaskFlowNet with recent techniques and datasets.

**Training iterations.** PWC-Net and IRR-PWC need large numbers of training iterations. At the same number of training iterations, IRR-PWC is consistently more accurate than PWC-Net. This is encouraging because we can perform an ablation study using fewer iterations and then use the best
setup to train the model using more iterations. One appealing feature of RAFT is its fast convergence, but we find that using more training iterations also improves RAFT. Note that 3.2M iterations for RAFT takes about 11 days while 6.2M iterations take PWC and IRR-PWC about 6 days to finish (using 6 P100 GPUs). It is interesting that all three models show no sign of over-fitting after so many iterations.

Other training details. We further test the effect of weight decay, random erasing and vertical flipping. As shown in Table 3, the training is robust to the hyperparameter settings for the weight decay, random erasing and vertical flipping.

### Table 2: Pre-training results for PWC-Net, IRR-PWC, RAFT and some recent methods. The metric for Sintel is average end-point error (AEPE) and F-all is the percentage of outliers averaged over all ground truth pixels. Lower is better for both AEPE and F-all. “-” means the same as the row above. C+T stands for the FlyingChairs and FlyingThings3D dataset schedule. Gradient clipping (GC), OneCycle learning rate, AutoFlow and longer training improve all three models consistently.

| Model          | Dataset  | GC   | LR     | Iter   | Sintel clean | Sintel final | KITTI clean | KITTI final | AEPE |
|----------------|----------|------|--------|--------|--------------|--------------|-------------|-------------|------|
| PWC-Net        | FlyingChairs | X    | Piecewise | 1.2M   | 3.89         | 4.79         | 42.81%      | 13.59       |
| -              |          |      |        | 3.2M   | 2.99         | 4.21         | 38.49%      | 10.7        |
| -              |          |      |        | 6.2M   | 2.10         | 2.81         | 16.29%      | 5.55        |
| IRR-PWC        | FlyingChairs | X    | Piecewise | 1.2M   | 4.3          | 5.09         | 44.06%      | 15.5        |
| -              |          |      |        |        | 3.01         | 4.11         | 26.95%      | 9.01        |
| RAFT           | FlyingChairs | X    | Piecewise | 0.2M   | 2.64         | 4.04         | 32.52%      | 10.01       |
| -              |          |      |        |        | 2.57         | 3.36         | 19.92%      | 5.96        |
| -              |          |      |        |        | 2.44         | 3.20         | 17.95%      | 5.49        |
| VCN            | C+T      | X    | Piecewise | 0.22M  | 2.21         | 3.62         | 25.10%      | 8.36        |
| MaskFlowNet    |          |      |        |        | 2.25         | 3.61         | 23.14%      | -           |
| Flow1D         |          |      |        |        | 1.98         | 3.27         | 22.95%      | 6.69        |
| SeparableFlow  |          |      |        |        | 1.30         | 2.59         | 15.90%      | 4.60        |

Table 3: More ablation studies on pre-training PWC-Net using 1.2M training steps. Default settings are underlined. Pre-training is robust to moderate variations on the parameters settings for these training details.

### Recipes for Pre-training. Using AutoFlow, gradient clipping, the OneCycle learning rate and long training consistently improves the pre-training results for PWC-Net, IRR-PWC and RAFT. It is
feasible to use short training to evaluate design choices and then use longer training times for the best performance.

### 3.3 Fine-tuning

To analyze fine-tuning, we use the training/validation split for Sintel proposed in Lv et al. [27], where the training and validation sets share different motion distributions (Fig. 3), and the training/validation split for KITTI proposed in Yang and Ramanan [53]. We follow [41] and use five datasets, Sintel [5] (0.4), KITTI [11] (0.2), VIPER [35] (0.2), HD1K [22] (0.08), and FlyingThings3D [28] (0.12), where the number indicates the sampling probability. We perform the ablation study on PWC-Net, and then apply the selected training protocol to IRR-PWC and RAFT.

| Model     | Data   | Init | Ft | Sintel | KITTI 2015 |
|-----------|--------|------|----|--------|------------|
|           |        |      |    | Training |            | Validation |
|           |        |      |    | clean   | final      | F-all      |
|           |        |      |    | AEPE    |            | AEPE       |
|           |        |      |    | clean   | final      | AEPE       |
|           |        |      |    | final   |            |            |
| PWC-Net   | SKHTV  | 1.2M | 1.2M | (1.04) | (1.45)     | 3.58       |
|           |        |      |    |         |            | 3.88       |
|           |        |      |    |         |            | (5.58%)    |
|           |        |      |    |         |            | (1.44)     |
|           |        |      |    |         |            | 6.23%      |
|           |        |      |    |         |            | 1.92       |
|           |        | -    | 3.2M | -       | (1.05)     | 2.95       |
|           |        |      |    |         | (1.55)     | 3.61       |
|           |        |      |    |         | (5.44%)    | (1.40)     |
|           |        |      |    |         | (1.61)     | 6.13%      |
|           |        |      |    |         | (1.80)     |            |
|           |        | No GC| -    | -       | (0.97)     | 3.09       |
|           |        |      |    |         | (1.42)     | 3.65       |
|           |        |      |    |         | (4.99%)    | (1.31)     |
|           |        |      |    |         | (5.61%)    | 1.62       |
|           |        | Piecewise| - | - | (1.08) | (1.62) | 3.32 |
|           |        |      |    |         | 3.77       |
|           |        |      |    |         | (5.49%)    |
|           |        |      |    |         | (1.42)     |
|           |        |      |    |         | 5.90%      |
|           |        |      |    |         | 1.78       |
| PWC-Net   | SKHTV  | 6.2M | 0M  | 1.78    | 2.55       |
|           |        |      |    |         | 3.33       |
|           |        |      |    |         | 3.83       |
|           |        |      |    |         | 16.50%     |
|           |        |      |    |         | 5.58       |
|           |        |      |    |         | 15.45%     |
|           |        |      |    |         | 5.44       |
| -         |        | -    | 6.2M | -       | (0.74)     |
|           |        |      |    |         | (1.08)     |
|           |        |      |    |         | 2.79       |
|           |        |      |    |         | 3.52       |
|           |        |      |    |         | (3.96%)    |
|           |        |      |    |         | (1.08)     |
|           |        |      |    |         | 4.76%      |
|           |        |      |    |         | 1.52       |
| -         |        | +A   | -    | -       | (0.80)     |
|           |        |      |    |         | (1.19)     |
|           |        |      |    |         | 2.76       |
|           |        |      |    |         | 3.25       |
|           |        |      |    |         | (4.10%)    |
|           |        |      |    |         | (1.12)     |
|           |        |      |    |         | 4.89%      |
|           |        |      |    |         | 1.57       |
| IRR-PWC   | SKHTV  | 6.2M | 0M  | 1.58    | 2.49       |
|           |        |      |    |         | 3.27       |
|           |        |      |    |         | 3.79       |
|           |        |      |    |         | 15.4%      |
|           |        |      |    |         | 5.05       |
|           |        |      |    |         | 14.3%      |
|           |        |      |    |         | 5.02       |
| -         |        | +A   | -    | -       | (0.98)     |
|           |        |      |    |         | (1.47)     |
|           |        |      |    |         | 2.85       |
|           |        |      |    |         | 3.50       |
|           |        |      |    |         | (4.52%)    |
|           |        |      |    |         | (1.21)     |
|           |        |      |    |         | 5.37%      |
|           |        |      |    |         | 1.59       |
| RAFT      | SKHTV  | 3.2M | 0M  | 1.40    | 2.31       |
|           |        |      |    |         | 2.88       |
|           |        |      |    |         | 3.38       |
|           |        |      |    |         | 15.57%     |
|           |        |      |    |         | 4.19       |
|           |        |      |    |         | 12.74%     |
|           |        |      |    |         | 4.13       |
| -         |        | -    | 1.2M | -       | (0.66)     |
|           |        |      |    |         | (1.14)     |
|           |        |      |    |         | 1.96       |
|           |        |      |    |         | 2.81       |
|           |        |      |    |         | (3.55%)    |
|           |        |      |    |         | (1.04)     |
|           |        |      |    |         | 3.96%      |
|           |        |      |    |         | 1.41       |
| -         |        | +A   | -    | -       | (0.74)     |
|           |        |      |    |         | (1.15)     |
|           |        |      |    |         | 2.00       |
|           |        |      |    |         | 2.76       |
|           |        |      |    |         | (3.86%)    |
|           |        |      |    |         | (1.09)     |
|           |        |      |    |         | 4.08%      |
|           |        |      |    |         | 1.39       |

Table 4: **Ablation study** on fine-tuning on Sintel and KITTI using the training/validation split for Sintel from [27] and for KITTI from [53]. GC stands for gradient clipping and () indicates training errors. OM for fine-tuning means that no fine-tuning has been done (initialization). S,K,H,T,V and A denote Sintel, KITTI, FlyingThings3D, HD1K, VIPER and AutoFlow datasets, respectively. Better initialization, more training steps and adding AutoFlow improve the performance.

**Training techniques.** Table 4 summarizes the results of the ablation study on PWC-Net. Better initialization tends to lead to better fine-tuning results, especially on the KITTI dataset. For the same initialization, longer training yields more accurate results on the held-out validation set. Removing gradient clipping results in a significant performance drop on the validation sets, and switching from the OneCycle to the piecewise learning rate results in moderate performance degradation too. We further experiment with adding the AutoFlow data to the fine-tuning process, and observe improvements for both PWC-Net and IRR-PWC on the Sintel validation set, and a small drop in performance on the KITTI validation set. Adding AutoFlow yields just a small improvement for RAFT on Sintel (we discuss this result again below with the in-distribution fine-tuning experiment).

**Model comparison.** Among the three models, RAFT has the best accuracy on the validation set. The initialization of RAFT is almost as accurate as the fine-tuned PWC-Net on the Sintel.final validation set using the training/validation split [27]. While IRR-PWC has higher training errors on Sintel than PWC-Net, the validation errors of the two models are similar. IRR-PWC has slightly worse performance on the KITTI validation set than PWC-Net.

**In-distribution fine-tuning.** The training and validation subsets for Sintel proposed by Lv et al. [27] have different motion distributions; the validation set has more middle-to-large range motion, as shown in Fig. 3. To examine the performance of fine-tuning when the training and validation sets have similar distributions, we perform fine-tuning experiments using another split by [53]. As summarized in Table 5, PWC-Net has lower errors than RAFT on the Sintel validation set. As shown in Fig. 3, both the training and validation sets by [53] concentrate on small motions, suggesting that RAFT is good at generalization to out-of-distribution large motion for the Lv et al. split. This generalization behavior likely explains why adding AutoFlow [41] does not significantly help RAFT.
Figure 3: Motion distributions for the Lv et al. [27] (left) and Yang and Ramanan [53] (right) training/validation splits. There is a mismatch between training and validation distributions for the Lv split, making it suitable for out-of-distribution fine-tuning test, while the other split is more suitable for in-distribution test.

in the experiment above. The result also suggests that PWC-Net may be a good option for applications dealing with small motions, e.g., the hole between the cart and the man in the front in Fig. 1.

Table 5: In-distribution fine-tuning using the training/validation split [53] for Sintel. The training and validation sets share similar motion distributions (c.f. Fig. 3).

|                      | Training clean | Validation clean | Validation final | F-all AEPE | F-all AEPE | F-all AEPE |
|----------------------|----------------|------------------|------------------|------------|------------|------------|
| PWC-Net              | 2.06           | 2.67             | 2.24             | 3.23       | 16.50%     | 5.58       |
| PWC-Net-ft           | (1.30)         | (1.67)           | 1.18             | 1.74       | (4.21%)    | (1.14)     |
| IRR-PWC              | 1.87           | 2.53             | 2.09             | 3.44       | 15.4%      | 5.05       |
| IRR-PWC-ft           | (1.34)         | (1.88)           | 1.55             | 2.31       | (4.94%)    | (1.29)     |
| RAFT                 | 1.74           | 2.24             | 1.74             | 2.91       | 13.57%     | 4.19       |
| RAFT-ft              | (1.14)         | (1.70)           | 1.37             | 2.14       | (5.06%)    | (1.61)     |

Recipes for Fine-tuning. Using better initialization and long training times helps fine-tuning. Both gradient clipping and the OneCycle learning rate help fine-tuning. Adding AutoFlow may help with generalization of the models.

3.4 Benchmark Results

We next apply the fine-tuning protocols above, with the full training sets from KITTI and Sintel, and then test the fine-tuned models on the public test sets. Table 1 summarizes the 2-frame results of previously published PWC-Net, IRR-PWC, and RAFT, our newly trained models, and several recent methods.

MPI Sintel. Our newly trained PWC-Net and IRR-PWC are substantially better than the respective, published models, with up to a 1 pixel reduction in average end-point error (AEPE) on the Sintel benchmark. As shown in Fig. 4, the newly trained PWC-Net can much better recover fine motion details than the published one [43]. The newly trained PWC-Net and IRR-PWC are even more accurate than some recent models [53, 60, 52] on the more challenging final pass, while being about 3× faster during inference.

Our newly trained RAFT is moderately better than the published RAFT [41, 45]. Among all published 2-frame methods it is only less accurate than SeparableFlow [58] while being more than 2× faster in inference. Figure 5 visually compares SeparableFlow and our newly trained RAFT on two challenging sequences from Sintel test. The newly trained RAFT makes a larger error on “Ambush_1” under heavy snow, but it correctly predicts the motion of the dragon and the background on “Market_4”. To some degree, these comparisons with recent methods compare the effect of innovations on architecture with training techniques, suggesting that there may be large gains for innovations on training techniques.
Figure 4: Representative visual results on KITTI and Sintel test sets by the original and our newly trained PWC-Net (both fine-tuned). Our newly trained PWC-Net can better recover fine details, e.g., the traffic sign (top), the stop sign and car (middle) and the small birds and the dragon’s right wing (green is correct, bottom).

Figure 5: Visual comparison on two challenging sequences from the Sintel test set. All 2-frame methods make large errors due to heavy snow on “Ambush_1”, while RAFT models have larger errors. For the fast moving dragon under motion blur in “Market_4”, the newly trained RAFT can better resolve the foreground motion from the background than SeparableFlow and the previously trained RAFT.

KITTI 2015. The newly trained PWC-Net and IRR are substantially better than the respective, published models, with more than 2 percent reduction in average outlier percentage (Fl-all) on the KITTI 2015 benchmark. Both are also more accurate than some more recent models. Our newly trained RAFT achieves a lower Fl-all than all published optical flow methods at the time of writing. While beyond the scope of this paper, we expect that our improved training protocols would result in improvements for other recent models as well.

3.5 Higher-resolution Input, Inference Time and Memory

We perform qualitative evaluations on 2K and 4K resolution inputs from Davis. Figure shows results for the newly trained models on three 1920×1080 resolution DAVIS samples. For each sample, all models produce similarly high quality optical flow fields. With thin structures, like the region
surrounding the racquet’s frame (top row), RAFT and PWC-Net create sharper motion boundaries. In Fig. 7, we also present optical flow results for the newly trained IRR-PWC and PWC-Net on 4K DAVIS samples. Overall, the optical flows from both methods are quite comparable, with the IRR-PWC showing slightly better motion smoothness on the jumping dog (top row in Fig. 7).

Table 6 presents a comparison of inference times and memory consumption on an NVIDIA V100 GPU. To account for initial kernel loading, we report the average of 100 runs. For each model, we test three spatial sizes: 1024×448 (1K), 1920×1080 (Full HD/2K), and 3840×2160 (4K). PWC-Net and IRR-PWC show comparable inference time. RAFT, in contrast, is 4.3× and 14.4× slower in 1K and 2K, respectively. In terms of memory, PWC-Net and IRR-PWC, again, show comparable performance. The increase in memory usage from 1K to 2K is almost linear for PWC-Net and IRR-PWC. On the other hand, RAFT uses more memory. Its footprint grows almost quadratically, by 3.8×, from 1K to 2K, and at 4K resolution, RAFT leads to out-of-memory (OOM).

Table 6

| Model   | Spatial Size | Inference Time (s) | Memory Usage (GB) |
|---------|--------------|--------------------|-------------------|
| PWC-Net | 1024×448     | 2.3                | 1.0               |
| IRR-PWC | 1024×448     | 2.4                | 1.0               |
| RAFT    | 1024×448     | 9.6                | 5.7               |

Figure 6: Visual results on Davis 2K. All three newly trained models produce similar high-quality results, while RAFT and PWC-Net better recover fine motion details, such as the hole in the racquet (top). Please enlarge and view on screen.

Figure 7: Visual results on Davis 4K from our newly trained models. We show only PWC-Net and IRR-PWC results since RAFT runs out of memory on the 16GB GPU.

3.6 Discussion

What makes RAFT better than PWC-Net? Our results show that several factors contribute to the performance gap between the published RAFT (5.10% Fl-all on KITTI 2015, see Table 1) and
PWC-Net (7.72%) models, including training techniques, datasets and architecture innovations. Recent training techniques and datasets significantly improve PWC-Net (5.54%) and IRR-PWC (5.73%). The newly trained models are competitive with published RAFT (5.10%) performance while maintaining their advantages in speed and memory requirements during inference. These insights also yield a newly trained RAFT model that is more accurate than all published methods on KITTI 2015 (4.31%). We conclude that innovations on training techniques and datasets are another fruitful path to performance gains, for both old and new optical flow models. After compensating for the differences in training techniques and datasets, we can identify the true performance gap between PWC-Net and RAFT that is solely due to architecture innovations (5.54% vs. 4.31% Fl-all on KITTI 2015). Future work should examine which specific architecture elements of RAFT are critical, and whether they may be transferable to other models.

No model to rule all. Our study also shows that there are several factors to consider when choosing an optical flow model, including flow accuracy, training time, inference time, memory cost and application scenarios. RAFT has the highest accuracy and faster convergence in training, but is slower at test time and has a high memory footprint. PWC-Net and IRR-PWC are more appealing for applications that require fast inference, low memory cost and high-resolution input. PWC-Net may be suitable for applications with small motions. Every model entails trade-offs between different requirements; no single model is superior on all metrics. Thus, researchers may wish to focus on specific metrics for improvement, thereby providing practitioners with more options.

4 Conclusions

We have evaluated three prominent optical flow models with improved training protocols and observed surprising and significant performance gains. The newly trained PWC-Net and IRR-PWC are more accurate than the more recent Flow1D model on KITTI 2015, while being about 3× faster during inference. Our newly trained RAFT is more accurate than all published optical flow methods on KITTI 2015. These results demonstrate the benefits of decoupling the contributions of model architectures, training techniques, and datasets to understand the sources of performance gains.

References

[1] S Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. IJCV, 2011.

[2] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3703–3712, 2019.

[3] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. IJCV, 1994.

[4] Irwan Bello, William Fedus, Xianzhi Du, Ekin Dogus Cubuk, Aravind Srinivas, Tsung-Yi Lin, Jonathon Shlens, and Barret Zoph. Revisiting resnets: Improved training and scaling strategies. Advances in Neural Information Processing Systems, 34, 2021.

[5] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In Proc. ECCV, 2012.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In CVPR, pages 248–255. Ieee, 2009.
[7] Abdelaziz Djelouah, Joaquim Campos, Simone Schaub-Meyer, and Christopher Schroers. Neural inter-frame compression for video coding. In CVPR, pages 6421–6429, 2019.

[8] 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. arXiv preprint arXiv:2010.11929, 2020.

[9] Alexey Dosovitskiy, Philipp Fischery, Eddy Ilg, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, Thomas Brox, et al. FlowNet: Learning optical flow with convolutional networks. In Proc. ICCV, 2015.

[10] Lijie Fan, Wenbing Huang, Chuang Gan, Stefano Ermon, Boqing Gong, and Junzhou Huang. End-to-end learning of motion representation for video understanding. In Proc. CVPR, 2018.

[11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In Proc. CVPR, pages 3354–3361. IEEE, 2012.

[12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[14] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 558–567, 2019.

[15] Tak-Wai Hui, Xiaou Tang, and Chen Change Loy. Litelflownet: A lightweight convolutional neural network for optical flow estimation. In Proc. CVPR, 2018.

[16] Junhwa Hur and Stefan Roth. Iterative residual refinement for joint optical flow and occlusion estimation. In Proc. CVPR, pages 5754–5763, 2019.

[17] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In Proc. CVPR, 2017.

[18] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super SloMo: High quality estimation of multiple intermediate frames for video interpolation. In Proc. CVPR, 2018.

[19] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9772–9781, 2021.

[20] Rico Jonschkowski, Austin Stone, Jonathan T Barron, Ariel Gordon, Kurt Konolige, and Anelia Angelova. What matters in unsupervised optical flow. In Proc. ECCV, 2020.

[21] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Deep video inpainting. In CVPR, pages 5792–5801, 2019.

[22] Daniel Kondermann, Rahul Nair, Katrin Honauer, Karsten Krispin, Jonas Andrilis, Alexander Brock, Burkhard Gissemann, Mohsen Rahimimoghaddam, Sabine Hofmann, Claus Brenner, et al. The hci benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving. In CVPR Workshops, pages 19–28, 2016.

[23] Lahav Lipson, Zachary Teed, and Jia Deng. Raft-stereo: Multilevel recurrent field transforms for stereo matching. In 3DV, pages 218–227. IEEE, 2021.

[24] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selfflow: Self-supervised learning of optical flow. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.
[25] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. arXiv preprint arXiv:2201.03545, 2022.

[26] Ao Luo, Fan Yang, Kunming Luo, Xin Li, Haoqiang Fan, and Shuaicheng Liu. Learning optical flow with adaptive graph reasoning. arXiv preprint arXiv:2202.03857, 2022.

[27] Zhaoyang Lv, Kihwan Kim, Alejandro Troccoli, Deqing Sun, James Rehg, and Jan Kautz. Learning rigidity in dynamic scenes with a moving camera for 3d motion field estimation. In Proc. ECCV, 2018.

[28] Nikolaus Mayer, Eddy Ilg, Philipp Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In Proc. CVPR, 2016.

[29] Lukas Mehl, Cedric Beschle, Andrea Barth, and Andrés Bruhn. An anisotropic selection scheme for variational optical flow methods with order-adaptive regularisation. In International Conference on Scale Space and Variational Methods in Computer Vision, pages 140–152. Springer, 2021.

[30] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In AAAI, 2018.

[31] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In CVPR, 2016.

[32] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In Proc. CVPR, 2017.

[33] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In CVPR, pages 12240–12249, 2019.

[34] Stephan R Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In Proceedings of the IEEE International Conference on Computer Vision, pages 2213–2222, 2017.

[35] Stephan R. Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, pages 2232–2241, 2017.

[36] Hao Shi, Yifan Zhou, Kailun Yang, Xiaoting Yin, and Kaiwei Wang. Csflow: Learning optical flow via cross strip correlation for autonomous driving. arXiv preprint arXiv:2202.00909, 2022.

[37] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. arXiv preprint arXiv:2106.10270, 2021.

[38] Austin Stone, Daniel Maurer, Alper Ayvaci, Anelia Angelova, and Rico Jonschkowski. Smurf: Self-teaching multi-frame unsupervised raft with full-image warping. In CVPR, pages 3887–3896, 2021.

[39] Jonathan Stroud, David Ross, Chen Sun, Jia Deng, and Rahul Sukthankar. D3d: Distilled 3d networks for video action recognition. In CVPR, pages 625–634, 2020.

[40] Deqing Sun, Stefan Roth, and Michael J Black. Secrets of optical flow estimation and their principles. In CVPR, pages 2432–2439. IEEE, 2010.

[41] 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.

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

[43] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Models matter, so does training: An empirical study of cnns for optical flow estimation. IEEE TPAMI, 2019.
[44] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[45] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In Proc. ECCV, 2020.

[46] Zachary Teed and Jia Deng. Raft-3d: Scene flow using rigid-motion embeddings. In CVPR, pages 8375–8384, 2021.

[47] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[48] Zhexiong Wan, Yuxin Mao, and Yuchao Dai. Praflow_rvc: Pyramid recurrent all-pairs field transforms for optical flow estimation in robust vision challenge 2020. *arXiv preprint arXiv:2009.06360*, 2020.

[49] Jianyuan Wang, Yiran Zhong, Yuchao Dai, Kaihao Zhang, Pan Ji, and Hongdong Li. Displacement-invariant matching cost learning for accurate optical flow estimation. *Advances in Neural Information Processing Systems*, 33:15220–15231, 2020.

[50] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *arXiv preprint arXiv:2110.00476*, 2021.

[51] Taihong Xiao, Jinwei Yuan, Deqing Sun, Qifei Wang, Xin-Yu Zhang, Kehan Xu, and Ming-Hsuan Yang. Learnable cost volume using the cayley representation. In *ECCV*, pages 483–499. Springer, 2020.

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

[53] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. In *NeurIPS*, volume 32, pages 794–805, 2019.

[54] Guorun Yang, Hengshuang Zhao, Jianping Shi, Zhidong Deng, and Jiaya Jia. SegStereo: Exploiting semantic information for disparity estimation. In *Proc. ECCV*, 2018.

[55] Zhichao Yin, Trevor Darrell, and Fisher Yu. Hierarchical discrete distribution decomposition for match density estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[56] 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*, pages 3–10. Springer, 2016.

[57] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *DAGM*, 2007.

[58] Feihu Zhang, Oliver J Woodford, Victor Adrian Prisacariu, and Philip HS Torr. Separable flow: Learning motion cost volumes for optical flow estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10807–10817, 2021.

[59] Hang Zhao, Chuang Gan, Wei-Chiu Ma, and Antonio Torralba. The sound of motions. In *CVPR*, pages 1735–1744, 2019.

[60] Shengyu Zhao, Yilun Sheng, Yue Dong, Eric I-Chao Chang, and Yan Xu. Maskflownet: Asymmetric feature matching with learnable occlusion mask. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[61] Xiaoqi Zhao, Youwei Pang, Lihe Zhang, Huchuan Lu, and Lei Zhang. Suppress and balance: A simple gated network for salient object detection. In *ECCV*, pages 35–51. Springer, 2020.
A More Visual Comparisons

Here we provide more visual examples to more comprehensively evaluate these models visually. Throughout the appendix, we add “-it” to each method to denote our newly trained model, where “it” stands for improved training.

A.1 PWC-it, IRR-it, and RAFT-it (down-up) on Davis 4K

In this subsection, we include 5 examples of our improved training models on Davis 4K images: Figures 8, 9, 10, 11, and 12. Note that due to memory constraints, RAFT-it requires the input to be downsampled and then the output flow to be upsampled to 4K.

Figure 8: PWC-it, IRR-it, and RAFT-it (down-up) on Davis 4K images. Note that RAFT-it requires the input images to be downsampled due to memory requirements and then the flow upscaled. Note the higher level of detail in IRR-it and PWC-Net-it: the clothes on the person on the left.

A.2 PWC-it, IRR-it, and RAFT-it on Davis Full HD

In this subsection, we include 5 examples of our improved training models on Davis 2K images: Figures 13, 14, 15, 16, and 17.
Figure 9: PWC-it, IRR-it, and RAFT-it (down-up) on Davis 4K images. Note that RAFT-it requires the input images to be downsampled due to memory requirements and then the flow upsampled. Note the higher level of detail in IRR-it and PWC-Net-it: the biker’s brim, the pant leg, etc.

A.3 Old vs New for PWC and RAFT on Davis 448x864

In this subsection, we include 8 examples of two-frame optical flow from PWC-original, RAFT-original, PWC-it, and RAFT-it evaluated on Davis images with resolution 448 by 864: Figures 18 and 19.

A.4 Old vs New for PWC and RAFT on Viper 1080x1920

In this subsection, we include 4 examples of two-frame optical flow from PWC-original, RAFT-original, PWC-it, and RAFT-it evaluated on Viper validation images with resolution 1080 by 1920: Figures 20.
Figure 10: PWC-it, IRR-it, and RAFT-it (down-up) on Davis 4K images. Note that RAFT-it requires the input images to be downsampled due to memory requirements and then the flow upsampled. Note the higher level of detail in IRR-it and PWC-Net-it: the smoke stack in the front of the tractor and the texture on the wheel.
Figure 11: PWC-it, IRR-it, and RAFT-it (down-up) on Davis 4K images. Note that RAFT-it requires the input images to be downsampled due to memory requirements and then the flow upsampled. Note the higher level of detail in IRR-it and PWC-Net-it: plant in front of the front lion.
Figure 12: PWC-it, IRR-it, and RAFT-it (down-up) on Davis 4K images. Note that RAFT-it requires the input images to be downsampled due to memory requirements and then the flow upscaled. Note the higher level of detail in IRR-it and PWC-Net-it: the plants in the foreground.
Figure 13: PWC-it, IRR-it, and RAFT-it on Davis 2K images.
Figure 14: PWC-it, IRR-it, and RAFT-it on Davis 2K images.
Figure 15: PWC-it, IRR-it, and RAFT-it on Davis 2K images.
Figure 16: PWC-it, IRR-it, and RAFT-it on Davis 2K images.
Figure 17: PWC-it, IRR-it, and RAFT-it on Davis 2K images.
Figure 18: PWC-orig, RAFT-orig vs PWC-it, RAFT-it on Davis 448x864
Figure 19: PWC-orig, RAFT-orig vs PWC-it, RAFT-it on Davis 448x864
Figure 20: PWC-orig, RAFT-orig vs PWC-it, RAFT-it on Viper 1080x1920