Disentangling Architecture and Training for Optical Flow

Deqing Sun[⋆,†] Charles Herrmann[⋆] Fitsum Reda
Michael Rubinstein David J. Fleet William T. Freeman

Google Research

Abstract. How important are training details and datasets to recent optical flow architectures like RAFT? And do they generalize? To explore these questions, rather than develop a new architecture, we revisit three prominent architectures, 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 show surprisingly large improvements, up to 30% versus original published results on Sintel and KITTI 2015 benchmarks. Our newly trained RAFT obtains an Fl-all score of 4.31% on KITTI 2015 and an avg. rank of 1.7 for end-point error on Middlebury. Our results demonstrate the benefits of separating the contributions of architectures, training techniques and datasets when analyzing performance gains of optical flow methods. Our source code is available at `https://autoflow-google.github.io`.

Keywords: Optical Flow; Architecture; Training; Evaluation

1 Introduction

The field of optical flow has witnessed rapid progress in recent years, driven largely by deep learning. FlowNet [10] first demonstrated the potential of deep learning for optical flow, while PWC-Net [45] was the first model to eclipse classical flow techniques. The widely-acclaimed RAFT model [48] 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 architecture for optical flow [20,28,31,38,51,54,55,62] and related tasks [24,49].

The success of RAFT has been attributed primarily to its novel architecture, including its multi-scale all-pairs cost volume, its recurrent update operator, and its up-sampling module. Meanwhile, 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 architectures?

We begin by revisiting the 2018 PWC-Net [45], and investigate the impact of datasets and training techniques for both pre-training and fine-tuning. We show

[⋆] Equal technical contribution, [†] project lead.

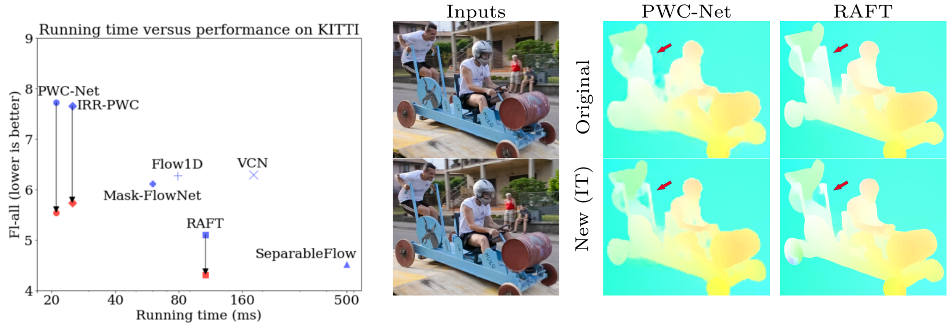

Fig. 1. Left: **Large improvements with newly trained PWC-Net, IRR-PWC and RAFT** (left: originally published results in blue; results of our newly trained models in red). Right: Visual comparison on a Davis sequence between the original [46] and our newly trained PWC-Net-it and RAFT-it, shows improved flow details, e.g. the hole between the cart and the person at the back. The newly trained PWC-Net-it recovers the hole between the cart and the front person better than RAFT.

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 [55,64] 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, which obtains competitive results on Sintel, KITTI, and VIPER while setting a new state of the art on Middlebury. We denote architectures trained with this new training by adding "-it" after the architecture name; for example, our newly trained RAFT will be abbreviated as RAFT-it.

We make the following contributions:

- We show that newly trained PWC-Net (PWC-Net-it), 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 (RAFT-it) 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.

Method	Sintel.clean	Sintel.final	KITTI	Running time
PWC-Net [45]	3.86	5.13	9.60%	30ms*
PWC-Net+ [46]	3.45	4.60	7.72%	30ms*
PWC-Net-it (Ours)	2.31	3.69	5.54%	**21ms**
IRR-PWC [17]	3.84	4.58	7.65%	180ms*
IRR-PWC-it (Ours)	2.19	3.55	5.73%	25ms
RAFT [48]	1.94	3.18	5.10%	94ms*
RAFT-A [44]	2.01	3.14	4.78%	107ms
RAFT-it (Ours)	1.55	2.90	**4.31%**	107ms
HD^3[58]	4.79	4.67	6.55%	100ms*
VCN [57]	2.81	4.40	6.30%	180ms*
Mask-FlowNet [64]	2.52	4.14	6.11%	60ms*
DICL [52]	2.12	3.44	6.31%	-
Flow1D [55]	2.24	3.81	6.27%	79ms*
RAFT+AOIR [31]	1.85	3.17	5.07%	10^{4}ms*
CSFlow [38]	1.63	3.03	5.00%	200ms*
SeparableFlow [62]	**1.50**	**2.67**	4.51%	250ms*

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 448×1024 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, PWC-Net-it and IRR-PWC-it are more accurate than some recent methods [55,56] on KITTI 2015 while being about 3× faster in inference.

2 Previous Work

Deep models for optical flow. FlowNet [10] was the first model to demonstrate the potential of deep learning for optical flow, and inspired various new architectures. FlowNet2 [18] stacked basic models to improve model capacity and performance, while SpyNet [34] used an image pyramid and warping to build a compact model. PWC-Net [45] used classical optical flow principles (e.g., [3,43,47]) to build an effective model, which has since seen widespread use [2,8,19,22,35,41,63,65]. The concurrent LiteFlowNet [16] used similar ideas to build a lightweight network. TVNet [11] took a different approach with classical flow principles by unrolling the optimization iterations of the TV-L1 method [61].

Many architectures have used a pyramid structure. IRR-PWC [17] introduced iterative refinement, reusing the same flow decoder module at different pyramidal levels. VCN [56] used a 4D cost volume that is easily adapted to stereo and optical flow. HD^3 [58] modeled flow uncertainty hierarchically. MaskFlowNet [64] 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 [48], 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 [20,28,31,38,51,54,55,62]. 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 [55]. SeparableFlow used a non-local aggregation module for cost aggregation, yielding substantial performance gains [62].

Recent research on optical flow has focused on architectural innovations. Nevertheless, most new optical flow papers combine new architectures with changes in training procedures and datasets. As such, it can be hard to identify which factors are responsible for the performance gains. In this paper, we take a different approach, instead we examine the effects of different ingredients of modern training techniques and datasets, but with established architectures. The results and findings are surprising. Our newly trained PWC-Net and IRR-PWC are more accurate than Flow1D while being almost 3× faster in inference, and our newly trained RAFT is more accurate than all published optical flow methods on KITTI 2015 while being more than 2× faster in inference than the previous best SeparableFlow.

Datasets for optical flow. For pre-training the predominant dataset is FlyingChairs [10]. Ilg *et al.* [18] introduced a dataset schedule that uses FlyingChairs and FlyingThings3D [30] sequentially. This remains a standard way to pre-train models. Sun *et al.* [44] 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 [23] and VIPER [36], 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.* [18] found that using dataset scheduling can improve the pre-training results of FlowNetS and FlowNetC. Sun *et al.* [46] 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 [45] by using additional datasets. Sun *et al.* [44] 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 [21,26,32,25,40,60], focusing more on the loss than model architecture. UFlow [21] 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. FOAL introduces a meta learning approach for online adaptation [59].

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 [15] and [53] examined modern training techniques on ResNet-50 [14] and observed significant performance improvements on ImageNet [7], 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) [9] also led to a series of papers [39,50] 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.* [27] 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 architecture, 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. We perform ablations on 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 [17] 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 $\mathcal{O}((HW)^2 D)$, where D is the number of feature channels (constant, typically 256 for RAFT and ≤ 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 $\mathcal{O}(HW(H+W)D)$. By comparison, the memory needed for the cost volume in PWC-Net and IRR-PWC is $\mathcal{O}(HWD(2d+1)^2)$, 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

Typical training recipes. A typical training pipeline trains models first on the FlyingChairs dataset, followed by fine-tuning on the FlyingThings3D dataset, and then further fine-tuning using a mixture of datasets, including 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 [46], 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 summaries 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 [44].) We further perform an ablation study on several key design choices using PWC-Net, 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 these elements [17].

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 [33], and Middlebury [1] sequences. PWC-Net trained on AutoFlow better recovers fine motion details (top) and produces coherent motion for the foreground objects (bottom).

Gradient clipping. Gradient clipping is a heuristic to avoid cliff structures for recurrent neural networks [13]. 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. Gradient clipping also improves the performance

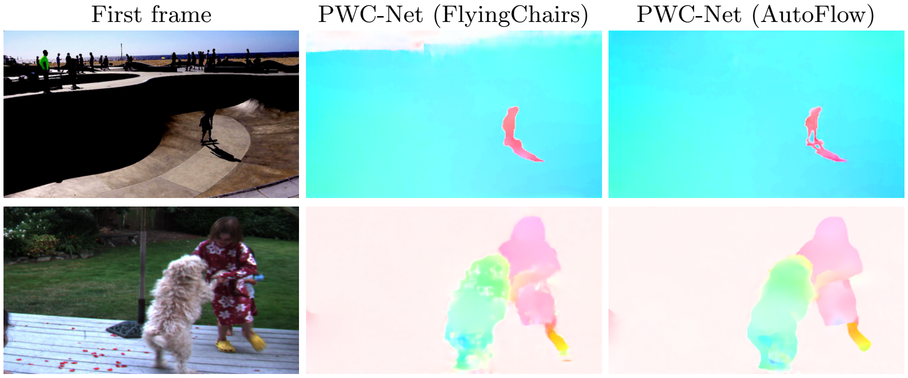

Fig. 2. Visual results of PWC-Net pre-trained using FlyingChairs and AutoFlow on Davis and Middlebury input images. PWC-Net trained using AutoFlow recovers fine details between the legs (top) and coherent motion for the girl and the dog (bottom).

of PWC-Net and IRR-PWC substantially and 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 robust to this parameter (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.

Model	Dataset	GC	LR	Iters	Sintel		KITTI	
					clean	final	F-all	AEPE
PWC-Net	FlyingChairs	✗	Piecewise	1.2M	3.89	4.79	42.81%	13.59
-	-	-	-	3.2M	2.99	4.21	38.49%	10.7
-	AutoFlow	✓	OneCycle	1.2M	2.43	3.05	18.74%	6.41
-	-	-	-	3.2M	2.17	2.91	17.25%	5.76
-	-	-	-	6.2M	2.10	2.81	16.29%	5.55
IRR-PWC	FlyingChairs	✗	Piecewise	1.2M	4.3	5.09	44.06%	15.5
-	AutoFlow	-	-	-	3.01	4.11	26.95%	9.01
-	-	✓	-	-	2.42	3.29	18.31%	6.31
-	-	-	OneCycle	-	2.24	2.93	17.87%	6.02
-	-	-	-	3.2M	2.06	2.85	15.55%	5.14
-	-	-	-	6.2M	1.93	2.76	15.20%	5.05
RAFT	FlyingChairs	✗	Piecewise	0.2M	2.64	4.04	32.52%	10.01
-	AutoFlow	-	-	-	2.57	3.36	19.92%	5.96
-	-	✓	-	-	2.44	3.20	17.95%	5.49
-	-	-	OneCycle	-	2.08	2.75	15.32%	4.66
-	-	-	-	0.8M	1.95	2.57	13.82%	4.23
-	-	-	-	3.2M	1.74	2.41	13.41%	4.18
VCN	C+T	✗	Piecewise	0.22M	2.21	3.62	25.10%	8.36
MaskFlowNet	-	-	-	1.7M	2.25	3.61	23.14%	-
Flow1D	-	✓	OneCycle	0.2M	1.98	3.27	22.95%	6.69
SeparableFlow	-	-	-	0.2M	1.30	2.59	15.90%	4.60

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.

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.* [29], where the sets have different motion distributions (Fig. 3), and the training/validation split for KITTI proposed in Yang and Ramanan [56]. We

Experiment	Parameter	Sintel		KITTI	
		clean	final	F-all	AEPE
Gradient clipping threshold	0.5	2.37	3.12	18.46%	6.14
	1.0	2.43	3.05	18.74%	6.41
	2.0	2.60	3.31	21.25%	7.73
Peak of OneCycle LR	0.1	2.38	3.04	17.35%	5.77
	0.2	2.43	3.05	18.74%	6.41
	0.3	2.35	3.08	19.39%	6.66
Weight decay	0	2.43	3.05	18.74%	6.41
	1e-8	2.31	3.09	18.07%	6.14
	1e-7	2.46	3.17	18.10%	6.17
Vertical flip probability	0	2.43	3.05	18.74%	6.41
	0.1	2.38	3.08	18.64%	6.14
Random erasing probability	0	2.43	3.05	18.74%	6.41
	0.5	2.46	3.13	17.39%	5.78

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.

follow [44] and use five datasets, Sintel [5] (0.4), KITTI [12] (0.2), VIPER [37] (0.2), HD1K [23] (0.08), and FlyingThings3D [30] (0.12), where the number indicates the sampling probability. We perform an ablation study on PWC-Net, and then apply the selected training protocol to IRR-PWC and RAFT.

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 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 [29]. 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.

				Sintel				KITTI 2015			
Model	Data	Init	Ft	Training		Validation		Training		Validation	
				clean	final	clean	final	F-all	AEPE	F-all	AEPE
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)	(1.55)	2.95	3.61	(5.44%)	(1.40)	6.13%	1.80
-	-	6.2M	-	(0.97)	(1.42)	3.09	3.65	(4.99%)	(1.31)	5.61%	1.62
No GC	-	-	-	(1.82)	(2.43)	4.51	4.98	(11.77%)	(3.06)	11.87%	3.79
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
-	-	-	6.2M	(0.98)	(1.47)	2.85	3.50	(4.52%)	(1.21)	5.37%	1.59
-	+A	-	-	(1.01)	(1.49)	2.64	3.28	(4.86%)	(1.29)	5.39%	1.56
RAFT	SKHTV	3.2M	0M	1.40	2.31	2.88	3.38	13.57%	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 [29] and for KITTI from [56]. GC stands for gradient clipping and () indicates training errors. 0M 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.

In-distribution fine-tuning. The training and validation subsets for Sintel proposed by Lv *et al.* [29] 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 [56]. 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 [56] 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 [44] does not significantly help RAFT 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.

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.

	Sintel				KITTI 2015			
	Training		Validation		Training		Validation	
	clean	final	clean	final	F-all	AEPE	F-all	AEPE
PWC-Net	2.06	2.67	2.24	3.23	16.50%	5.58	15.45%	5.44
PWC-Net-ft	(1.30)	(1.67)	1.18	1.74	(4.21%)	(1.14)	5.10%	1.51
IRR-PWC	1.87	2.53	2.09	3.44	15.4%	5.05	14.3%	5.02
IRR-PWC-ft	(1.34)	(1.88)	1.55	2.31	(4.94%)	(1.29)	5.42%	1.65
RAFT	1.74	2.24	1.74	2.91	13.57%	4.19	12.74%	4.13
RAFT-ft	(1.14)	(1.70)	1.37	2.14	(5.06%)	(1.61)	5.01%	1.40

Table 5. In-distribution fine-tuning using the training/validation split [56] for Sintel. The training and validation sets share similar motion distributions (*c.f.*Fig. 3).

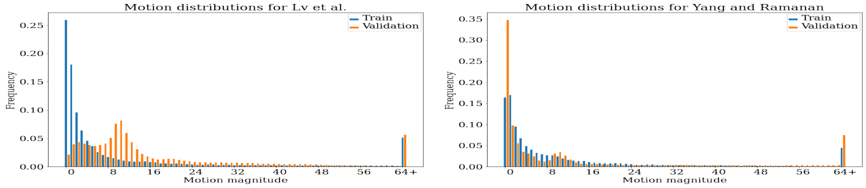

Fig. 3. Motion distributions for the Lv *et al.* [29] (left) and Yang and Ramanan [56] (right) training/validation splits. There is a mismatch between training and validation distributions for the Lv split, making it suitable for out-of-distribution fine-tuning test, while the other split is more suitable for in-distribution test.

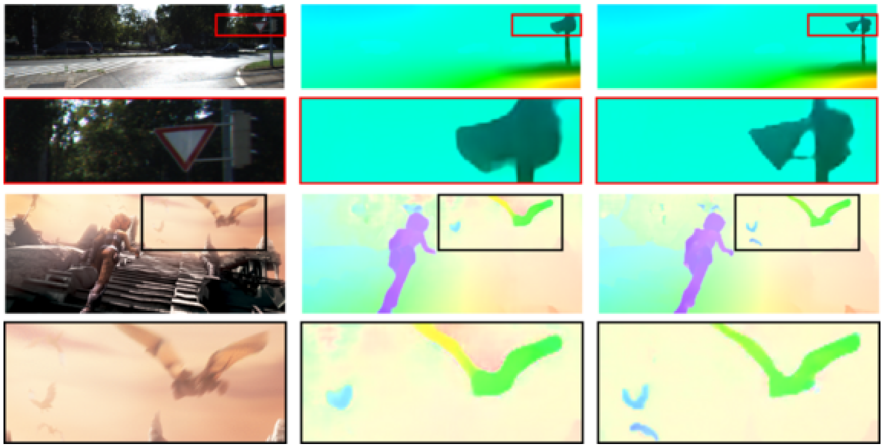

Fig. 4. Representative visual results on KITTI and Sintel test sets by the original [46] and our newly trained PWC-Net (both fine-tuned). Our newly trained PWC-Net can better recover fine details, *e.g.*, the traffic sign (top) and the small birds and the dragon's right wing (green is correct, bottom).

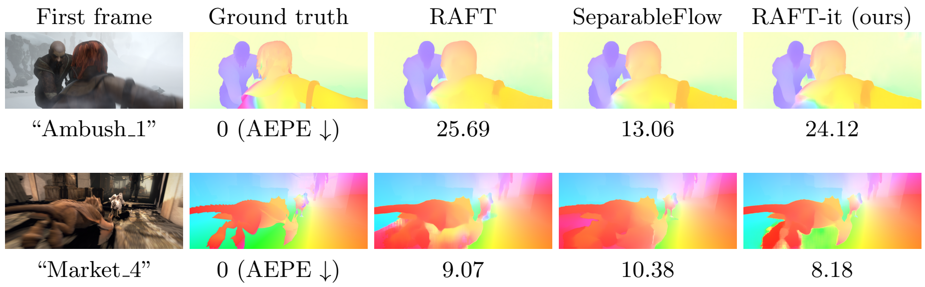

Fig. 5. Visual comparison on two challenging sequences from the Sintel test set. All 2-frame methods make large errors due to heavy snow on "Ambush_1", while RAFT models have larger errors. For the fast moving dragon under motion blur in "Market_4", the newly trained RAFT-it can better resolve the foreground motion from the background than SeparableFlow and the previously trained RAFT [44].

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-it and IRR-PWC-it 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, PWC-Net-it can much better recover fine motion details than the published one [46]. PWC-Net-it and IRR-PWC-it are even more accurate than some recent models [56,64,55] on the more challenging final pass, while being about 3× faster during inference.

Our newly trained RAFT-it is moderately better than the published RAFT [44,48]. Among all published 2-frame methods it is only less accurate than SeparableFlow [62] 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. RAFT-it 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.

KITTI 2015. The newly trained PWC-Net-it and IRR-it 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 [31,55,56,64].

Middlebury. At the time of writing, our newly trained RAFT-it is ranked first on Middlebury for both end-point and angular errors, with the avg. rank being 1.7 and 3.9, respectively. It is the first deep learning based approach to outperform traditional methods on Middlebury, such as NNF-Local [6] (avg. rank 5.8 and 7.4), which had been the top-performing method since 2013.

VIPER. Our newly trained RAFT-it obtains 73.6 for the mean weighted area under the curve (WAUC) over all conditions, *v.s.* 69.5 by RAFT_RVC [42].

3.5 Higher-resolution Input, Inference Time and Memory

We perform qualitative evaluations on 2K and 4K resolution inputs from Davis [33]. For 2K, all models produce similarly high quality flow fields, please see the supplementals for images. In Fig. 6, we present optical flow results for the newly trained IRR-PWC-it and PWC-Net-it on 4K DAVIS samples. Overall, the flows are comparable, with IRR-PWC-it showing slightly better motion smoothness on the jumping dog (top row in Fig. 6).

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).

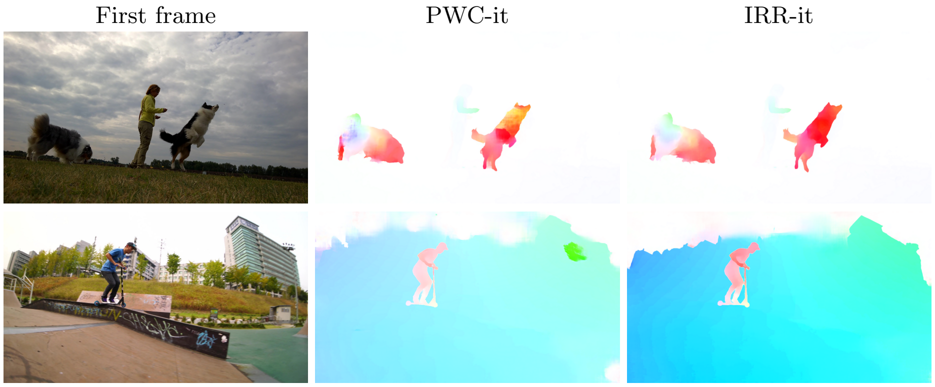

Fig. 6. Visual results on **Davis 4K**. We show only PWC-Net-it and IRR-PWC-it results since RAFT runs out of memory on the 16GB GPU.

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%) methods, including training

	Inference Time (msec)↓			Peak Memory (GB)↓		
	1024×448	Full HD	4K	1024 × 448	Full HD	4K
PWC-Net	20.61	28.77	63.31	1.478	2.886	7.610
IRR-PWC	24.71	33.67	57.59	1.435	2.902	8.578
RAFT	107.38	499.63	n/a	2.551	9.673	OOM

Table 6. Inference time and memory usage for 1024×448, Full HD (1920×1080) and 4K (3840×2160) frame sizes, averaged over 100 runs on an NVIDIA V100 GPU.

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-it model that sets a new state of the art on Middlebury at the time of writing. We conclude that innovations on training techniques and datasets are another fruitful path to performance gains, for both old and new optical flow architectures. 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 architectures with improved training protocols and observed surprising and significant performance gains. The newly trained PWC-Net-it and IRR-PWC-it are more accurate than the more recent Flow1D model on KITTI 2015, while being about 3× faster during inference. Our newly trained RAFT-it sets a new state of the art and is the first deep learning approach to outperform traditional methods on the Middlebury benchmark. 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. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. IJCV (2011) 6
2. Bao, W., Lai, W.S., Ma, C., Zhang, X., Gao, Z., Yang, M.H.: Depth-aware video frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3703–3712 (2019) 3
3. Barron, J., Fleet, D., Beauchemin, S.: Performance of optical flow techniques. IJCV (1994) 3
4. Bello, I., Fedus, W., Du, X., Cubuk, E.D., Srinivas, A., Lin, T.Y., Shlens, J., Zoph, B.: Revisiting resnets: Improved training and scaling strategies. Advances in Neural Information Processing Systems **34** (2021) 5
5. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: Proc. ECCV (2012) 9
6. Chen, Z., Jin, H., Lin, Z., Cohen, S., Wu, Y.: Large displacement optical flow from nearest neighbor fields. In: CVPR. pp. 2443–2450 (2013) 12
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255. Ieee (2009) 5
8. Djelouah, A., Campos, J., Schaub-Meyer, S., Schroers, C.: Neural inter-frame compression for video coding. In: CVPR. pp. 6421–6429 (2019) 3
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) 5
10. Dosovitskiy, A., Fischery, P., Ilg, E., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., Brox, T., et al.: FlowNet: Learning optical flow with convolutional networks. In: Proc. ICCV (2015) 1, 3, 4
11. Fan, L., Huang, W., Gan, C., Ermon, S., Gong, B., Huang, J.: End-to-end learning of motion representation for video understanding. In: Proc. CVPR (2018) 3
12. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Proc. CVPR. pp. 3354–3361. IEEE (2012) 9
13. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016) 6
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 5
15. He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., Li, M.: Bag of tricks for image classification with convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 558–567 (2019) 5
16. Hui, T.W., Tang, X., Change Loy, C.: Liteflownet: A lightweight convolutional neural network for optical flow estimation. In: Proc. CVPR (2018) 3
17. Hur, J., Roth, S.: Iterative residual refinement for joint optical flow and occlusion estimation. In: Proc. CVPR. pp. 5754–5763 (2019), github.com/visinf/irr/blob/master/models/pwcnet_irr.py 3, 5, 6
18. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: FlowNet 2.0: Evolution of optical flow estimation with deep networks. In: Proc. CVPR (2017) 3, 4
19. Jiang, H., Sun, D., Jampani, V., Yang, M.H., Learned-Miller, E., Kautz, J.: Super SloMo: High quality estimation of multiple intermediate frames for video interpolation. In: Proc. CVPR (2018) 3

20. Jiang, S., Campbell, D., Lu, Y., Li, H., Hartley, R.: Learning to estimate hidden motions with global motion aggregation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9772–9781 (2021) 1, 4
21. Jonschkowski, R., Stone, A., Barron, J.T., Gordon, A., Konolige, K., Angelova, A.: What matters in unsupervised optical flow. In: Proc. ECCV (2020) 4
22. Kim, D., Woo, S., Lee, J.Y., Kweon, I.S.: Deep video inpainting. In: CVPR. pp. 5792–5801 (2019) 3
23. Kondermann, D., Nair, R., Honauer, K., Krispin, K., Andrulis, J., Brock, A., Gussefeld, B., Rahimimoghaddam, M., Hofmann, S., Brenner, C., et al.: The hci benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving. In: CVPR Workshops. pp. 19–28 (2016) 4, 9
24. Lipson, L., Teed, Z., Deng, J.: Raft-stereo: Multilevel recurrent field transforms for stereo matching. In: 3DV. pp. 218–227. IEEE (2021) 1
25. Liu, L., Zhang, J., He, R., Liu, Y., Wang, Y., Tai, Y., Luo, D., Wang, C., Li, J., Huang, F.: Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In: CVPR. pp. 6489–6498 (2020) 4
26. Liu, P., Lyu, M., King, I., Xu, J.: Selflow: Self-supervised learning of optical flow. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019) 4
27. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. arXiv preprint arXiv:2201.03545 (2022) 5
28. Luo, A., Yang, F., Luo, K., Li, X., Fan, H., Liu, S.: Learning optical flow with adaptive graph reasoning. arXiv preprint arXiv:2202.03857 (2022) 1, 4
29. Lv, Z., Kim, K., Troccoli, A., Sun, D., Rehg, J., Kautz, J.: Learning rigidity in dynamic scenes with a moving camera for 3d motion field estimation. In: Proc. ECCV (2018) 8, 9, 10, 11
30. Mayer, N., Ilg, E., Häusser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Proc. CVPR (2016) 4, 9
31. Mehl, L., Beschle, C., Barth, A., Bruhn, A.: 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. pp. 140–152. Springer (2021) 1, 3, 4, 12
32. Meister, S., Hur, J., Roth, S.: Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In: AAAI (2018) 4
33. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: CVPR (2016) 6, 13
34. Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network. In: Proc. CVPR (2017) 3
35. Ranjan, A., Jampani, V., Balles, L., Kim, K., Sun, D., Wulff, J., Black, M.J.: Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In: CVPR. pp. 12240–12249 (2019) 3
36. Richter, S.R., Hayder, Z., Koltun, V.: Playing for benchmarks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2213–2222 (2017) 4
37. Richter, S.R., Hayder, Z., Koltun, V.: Playing for benchmarks. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. pp. 2232–2241 (2017). https://doi.org/10.1109/ICCV.2017.243, `https://doi.org/10.1109/ICCV.2017.243` 9

38. Shi, H., Zhou, Y., Yang, K., Yin, X., Wang, K.: Csflow: Learning optical flow via cross strip correlation for autonomous driving. arXiv preprint arXiv:2202.00909 (2022) 1, 3, 4
39. Steiner, A., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., Beyer, L.: How to train your vit? data, augmentation, and regularization in vision transformers. arXiv preprint arXiv:2106.10270 (2021) 5
40. Stone, A., Maurer, D., Ayvaci, A., Angelova, A., Jonschkowski, R.: Smurf: Self-teaching multi-frame unsupervised raft with full-image warping. In: CVPR. pp. 3887–3896 (2021) 4
41. Stroud, J., Ross, D., Sun, C., Deng, J., Sukthankar, R.: D3d: Distilled 3d networks for video action recognition. In: CVPR. pp. 625–634 (2020) 3
42. Sun, D., Herrmann, C., Jampani, V., Krainin, M., Cole, F., Stone, A., Jonschkowski, R., Zabih, R., Freeman, W.T., Liu, C.: TF-RAFT: A tensorflow implementation of raft. In: ECCV Robust Vision Challenge Workshop (2020) 13
43. Sun, D., Roth, S., Black, M.J.: Secrets of optical flow estimation and their principles. In: CVPR. pp. 2432–2439. IEEE (2010) 3
44. Sun, D., Vlasic, D., Herrmann, C., Jampani, V., Krainin, M., Chang, H., Zabih, R., Freeman, W.T., Liu, C.: Autoflow: Learning a better training set for optical flow. In: CVPR (2021) 3, 4, 6, 9, 10, 12
45. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: CVPR (June 2018) 1, 3, 4
46. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Models matter, so does training: An empirical study of cnns for optical flow estimation. IEEE TPAMI (2019) 2, 3, 4, 6, 11, 12
47. Szeliski, R.: Computer vision: algorithms and applications. Springer Science & Business Media (2010) 3
48. Teed, Z., Deng, J.: RAFT: Recurrent all-pairs field transforms for optical flow. In: Proc. ECCV (2020) 1, 3, 12
49. Teed, Z., Deng, J.: Raft-3d: Scene flow using rigid-motion embeddings. In: CVPR. pp. 8375–8384 (2021) 1
50. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning. pp. 10347–10357. PMLR (2021) 5
51. Wan, Z., Mao, Y., Dai, Y.: Praflow_rvc: Pyramid recurrent all-pairs field transforms for optical flow estimation in robust vision challenge 2020. arXiv preprint arXiv:2009.06360 (2020) 1, 4
52. Wang, J., Zhong, Y., Dai, Y., Zhang, K., Ji, P., Li, H.: Displacement-invariant matching cost learning for accurate optical flow estimation. Advances in Neural Information Processing Systems **33**, 15220–15231 (2020) 3
53. Wightman, R., Touvron, H., Jégou, H.: Resnet strikes back: An improved training procedure in timm. arXiv preprint arXiv:2110.00476 (2021) 5
54. Xiao, T., Yuan, J., Sun, D., Wang, Q., Zhang, X.Y., Xu, K., Yang, M.H.: Learnable cost volume using the cayley representation. In: ECCV. pp. 483–499. Springer (2020) 1, 4
55. Xu, H., Yang, J., Cai, J., Zhang, J., Tong, X.: High-resolution optical flow from 1d attention and correlation. In: ICCV (2021) 1, 2, 3, 4, 12
56. Yang, G., Ramanan, D.: Volumetric correspondence networks for optical flow. In: NeurIPS. vol. 32, pp. 794–805 (2019) 3, 8, 10, 11, 12
57. Yang, G., Zhao, H., Shi, J., Deng, Z., Jia, J.: SegStereo: Exploiting semantic information for disparity estimation. In: Proc. ECCV (2018) 3

58. Yin, Z., Darrell, T., Yu, F.: Hierarchical discrete distribution decomposition for match density estimation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019) 3
59. Yu, H., Sun, S., Yu, H., Chen, X., Shi, H., Huang, T.S., Chen, T.: Foal: Fast online adaptive learning for cardiac motion estimation. In: CVPR. pp. 4313–4323 (2020) 5
60. Yu, J.J., Harley, A.W., Derpanis, K.G.: Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In: ECCV. pp. 3–10. Springer (2016) 4
61. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime tv-l 1 optical flow. In: DAGM (2007) 3
62. Zhang, F., Woodford, O.J., Prisacariu, V.A., Torr, P.H.: Separable flow: Learning motion cost volumes for optical flow estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10807–10817 (2021) 1, 3, 4, 12
63. Zhao, H., Gan, C., Ma, W.C., Torralba, A.: The sound of motions. In: CVPR. pp. 1735–1744 (2019) 3
64. Zhao, S., Sheng, Y., Dong, Y., Chang, E.I.C., Xu, Y.: Maskflownet: Asymmetric feature matching with learnable occlusion mask. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 2, 3, 12
65. Zhao, X., Pang, Y., Zhang, L., Lu, H., Zhang, L.: Suppress and balance: A simple gated network for salient object detection. In: ECCV. pp. 35–51. Springer (2020) 3