

# What Matters in Unsupervised Optical Flow

Rico Jonschkowski<sup>1,2</sup>, Austin Stone<sup>1,2</sup>, Jonathan T. Barron<sup>2</sup>,  
Ariel Gordon<sup>1,2</sup>, Kurt Konolige<sup>1,2</sup>, and Anelia Angelova<sup>1,2</sup>

<sup>1</sup>Robotics at Google and <sup>2</sup>Google AI

{rjon,austinstone,barron,gariel,konolige,anelia}@google.com

**Abstract.** We systematically compare and analyze a set of key components in unsupervised optical flow to identify which photometric loss, occlusion handling, and smoothness regularization is most effective. Alongside this investigation we construct a number of novel improvements to unsupervised flow models, such as cost volume normalization, stopping the gradient at the occlusion mask, encouraging smoothness before up-sampling the flow field, and continual self-supervision with image resizing. By combining the results of our investigation with our improved model components, we are able to present a new unsupervised flow technique that significantly outperforms the previous unsupervised state-of-the-art and performs on par with supervised FlowNet2 on the KITTI 2015 dataset, while also being significantly simpler than related approaches.

## 1 Introduction

Optical flow is a key representation in computer vision that describes the pixel-level correspondence between two images. Since optical flow is useful for estimating motion, disparity, and semantic correspondence, improvements in optical flow directly benefit downstream tasks such as visual odometry, stereo depth estimation, and object tracking. The performance of optical flow techniques has recently seen dramatic improvements, due to the widespread adoption of deep learning. Because ground-truth labels for dense optical flow are difficult to obtain for real image pairs, supervised optical flow techniques are primarily trained using synthetic data [5]. Although models trained on synthetic data often generalize well to real images, there is an inherent mismatch between these two data sources that those approaches may struggle to overcome [17,28].

Though non-synthetic data for training *supervised* optical flow techniques is scarce, the data required to train an *unsupervised* model is abundant: all that training requires is unlabeled video, of which there are countless hours freely available on the internet. If an unsupervised approach could leverage this abundant and diverse real data, it would produce an optical flow model that does not suffer from any mismatch between its training data and its test data, and could presumably produce higher-quality results. The core assumption shared by unsupervised optical flow techniques is that an object’s appearance does not change as it moves, which allows these models to be trained using unlabeled video as follows: The model is used to estimate a flow field between two images,

that flow field is used to warp one image to match the other, and then the model weights are updated so as to minimize the difference between those two images – and to accommodate some form of regularization.

Although all unsupervised optical flow methods share this basic idea, their details vary greatly. In this work we systematically compare, improve, and integrate key components to further our understanding and provide a unified framework for unsupervised optical flow. Our contributions are:

1. We systematically compare key components of unsupervised optical flow, such as photometric losses, occlusion estimation techniques, self-supervision, and smoothness constraints, and we analyze the effect of other choices, such as pretraining, image resolution, data augmentation, and batch size.
2. We propose four improvements to these key components: cost volume normalization, gradient stopping for occlusion estimation, applying smoothness at the native flow resolution, and image resizing for self-supervision.
3. We integrate the best performing improved components in a unified framework for unsupervised optical flow (UFlow for short) that sets a new state of the art – even compared to substantially more complex methods that estimate flow from multiple frames or co-train flow with monocular or stereo depth estimation. To facilitate future research, our source code is available at <https://github.com/google-research/google-research/tree/master/uflow>.

## 2 Related Work

The motion between an object and a viewer causes apparent movement of brightness patterns in the image [7]. Optical flow techniques attempt to invert this relationship to recover a motion estimate [16]. Classical methods infer optical flow for a pair of images by minimizing a loss function that measures photometric consistency and smoothness [10,3,24]. Recent approaches reframe optical flow estimation as a learning problem in which a CNN-based model regresses from a pair of images to a flow field [5,11]. Some models incorporate ideas from earlier methods, such as cost volumes and coarse-to-fine warping [20,25,32]. These supervised approaches require representative training data with accurate optical flow labels. Though such data can be generated for rigid objects with known geometry [6,19], recovering this ground truth flow for arbitrary scenes is laborious, and requires approaches as unusual as manually painting scenes with textured fluorescent paint and imaging it under ultraviolet light [1]. Since such approaches scale poorly, supervised methods have mainly relied on synthetic data for training, and often for evaluation [2,4,5]. Synthesizing “good” training data (such that learned models generalize to real images) is itself a hard research problem, requiring careful consideration of scene content, camera motion, lens distortion, and sensor degradation [17].

Unsupervised approaches circumvent the need for labels by optimizing photometric consistency with some regularization [22,34], similar to the classical optimization-based methods mentioned above. Where traditional methods solve an optimization problem for each image pair, unsupervised learning jointly op-

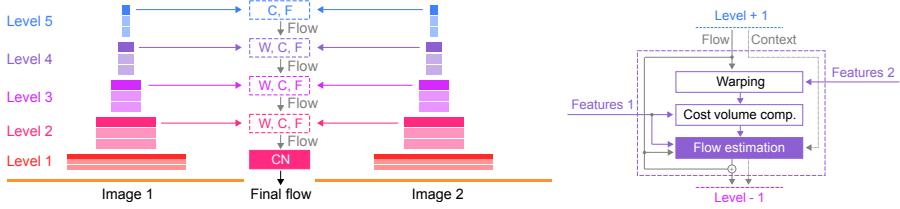
timizes an objective across all pairs in a dataset and learns a function that regresses a flow field from images. This approach has two advantages: 1) inference is fast because optimization is only performed during training, and 2) by jointly optimizing across the whole train set, information is shared across image pairs which can potentially improve performance. This unsupervised approach was extended to use edge-aware smoothness [30], a bi-directional Census loss [18], different forms of occlusion estimation [12,18,30], self-supervision [14,15], and estimation from multiple frames [12,15]. Other extensions introduced geometric reasoning through epipolar constraints [35] or by co-training optical flow with depth and ego-motion models from monocular [21,33,36] or stereo input [29].

These works have pushed the state of the art and generated a range of ideas for unsupervised optical flow. But since each of them evaluates a different *combination* of ideas, it is unclear how *individual* ideas compare to each other and which ideas combine well together. For example, the methods OAFlow [30] and DDFlow [14] use different photometric losses and different ways to mask occlusions, and OAFlow uses an edge-aware smoothness loss while DDFlow regularizes learning through self-supervision. DDFlow performs better than OAFlow, but does this mean that every component of DDFlow is better than every component of OAFlow? The ablation studies often presented in these papers show that each novel contribution of each work does indeed improve the performance of each individual model, but they do not provide a guarantee that each such contribution will always improve performance when added to any *other* model. Our work addresses this problem by systematically comparing and combining photometric losses (L1, Charbonnier [24], Census [14,18,35,36], and structural similarity [21,29,33]), different methods for occlusion estimation [3,30], first order and second order edge-aware smoothness [27], and self-supervision [14]. Our work also improves cost volume computation, occlusion estimation, smoothness, and self-supervision and integrates all components into an state of the art framework for unsupervised optical, while being simpler than many proposed methods to form a solid base for future work.

### 3 Preliminaries on Unsupervised Optical Flow

The task of estimating optical flow can be defined as follows: Given two color images  $I^{(1)}, I^{(2)} \in \mathbb{R}^{H \times W \times 3}$ , we want to estimate the flow field  $V^{(1)} \in \mathbb{R}^{H \times W \times 2}$ , which for each pixel in  $I^{(1)}$  denotes the relative position of its corresponding pixel in  $I^{(2)}$ . Note that optical flow is an asymmetric representation of pixel motion:  $V^{(1)}$  provides a flow vector for each pixel in  $I^{(1)}$ , but to find a mapping from image 2 back to image 1, one would need to estimate  $V^{(2)}$ .

In the context of unsupervised learning, we want to find a function  $V^{(1)} = f_{\theta}(I^{(1)}, I^{(2)})$  with parameters  $\theta$  learned from a set of image sequences  $D = \{(I^{(1)}, I^{(2)}, \dots, I^{(N)})\}$ . Because we lack ground truth flow, we must define a proxy objective  $\mathcal{L}(D, \theta)$ , such as photometric consistency between  $I^{(1)}$  and  $I^{(2)}$  after it has been warped according to some estimated  $V^{(1)}$ . To enforce photometric consistency only for pixels that can be reconstructed from the other



**Fig. 1.** Model overview. *Left:* Feature pyramids feed into a top-down flow estimation. *Right:* A zoomed in view on a “W, C, F” (warping, cost volume, flow estimation) block

image, we must also estimate an occlusion mask  $O^{(1)} \in \mathbb{R}^{H \times W}$ , for example based on the estimated forward and backward flow fields  $O^{(1)} = g(V^{(1)}, V^{(2)})$ .  $\mathcal{L}(\cdot)$  might also include other terms for, e.g. for smoothness or self-supervision. If  $\mathcal{L}(\cdot)$  is differentiable with respect to  $\theta$ , the parameters that minimize this loss  $\theta^* = \arg \min(\mathcal{L}(D, \theta))$  can be recovered using gradient-based optimization.

## 4 Key Components of Unsupervised Optical Flow

This section compares and improves key components of unsupervised optical flow. We will first discuss a model  $f_\theta(\cdot)$ , which we base on PWC-Net [25], and improve through *cost-volume normalization*. Then we go through different components of the objective function  $\mathcal{L}(\cdot)$ : occlusion-aware photometric consistency, smoothness, and self-supervision. Here, we propose improvements to each component: *stopping the gradient at the occlusion mask*, *computing smoothness at the native flow resolution*, and *image resizing for self-supervision*. We end this section by discussing data augmentation and optimization.

As shown in Fig. 1, our model feeds images  $I^{(1)}$  and  $I^{(2)}$  into a shared CNN that generates a feature pyramid, where features are used as input for warping (W), cost volume computation (C), and flow estimation (F). At each level  $\ell$ , the estimated flow  $V^{(1,\ell+1)}$  from the level above is upsampled, passed down to the lower level as  $\hat{V}^{(1,\ell+1)}$ , and then used to warp  $F^{(2,\ell)}$ , the features of image 2. The warped features  $w(F^{(2,\ell)}, \hat{V}^{(1,\ell)})$  together with  $F^{(1,\ell)}$  are used to compute a cost volume. The cost volume considers feature correlations for all pixels and all 81 combinations of shifting  $w(F^{(2,\ell)}, \hat{V}^{(1,\ell)})$  up to 4 pixels up/down and left/right. This results in a cost volume  $C^\ell \in \mathbb{R}^{\frac{W}{2^\ell} \times \frac{H}{2^\ell} \times 81}$  that describes how closely each pixel in  $F^{(1,\ell)}$  resembles the 81 pixels around its location in  $F^{(2,\ell)}$ . The cost volume, the features from image 1, the higher level flow and the *context* – the output of the second to last layer of the flow estimation network – are fed into a CNN that estimates a flow  $V^{(1,\ell)}$ . After a number of flow estimation levels, there is a final stage of flow refinement at level two in which the flow and context are fed into a *context network* (CN), which is a stack of dilated convolutions.

*Model Shrinking, Level Dropout and Cost Volume Normalization:* PWC-Net was designed for supervised learning of optical flow [25]. To deal with increased

memory requirements for unsupervised learning due to bi-directional losses, occlusion estimation, and self-supervision, we remove level six, use 32 channels in all levels, and add residual connections to all flow estimation modules (the “+” in the bottom right of Fig. 1). Additionally, we dropout residual flow estimation at all levels to further regularize learning, i.e. we randomly pass the resized and rescaled flow estimate from the level above directly to the level below.

Another difference when using this model for unsupervised rather than supervised learning is that unsupervised losses are typically only imposed on the final output (presumably because photometric consistency and other objectives work better at higher resolutions). But without supervised losses on intermediate flow predictions, the model has difficulty learning flow estimation at higher levels. We found that this is caused by very low values in the estimated cost volumes as a result of vanishing feature activations at higher levels.

We address this problem by *cost volume normalization*. Let us denote features for image  $i$  at level  $\ell$  as  $F^{(i,\ell)} \in \mathbb{R}^{\frac{H}{2^\ell} \times \frac{W}{2^\ell} \times d}$ . The cost volume between images 1 and 2 for all image locations  $(x, y)$  and all considered image shifts  $(u, v)$  is the inner product of the normalized features of the two images:

$$C_{x,y,u,v}^{(\ell)} = \sum_d \left( \frac{F_{x,y,d}^{(1,\ell)} - \mu^{(1,\ell)}}{\sigma^{(1,\ell)}} \right) \left( \frac{F_{x+u,y+v,d}^{(2,\ell)} - \mu^{(2,\ell)}}{\sigma^{(2,\ell)}} \right). \quad (1)$$

Where  $\mu^{(i,\ell)}$  and  $\sigma^{(i,\ell)}$  are the sample mean and standard deviation of  $F^{(i,\ell)}$  over its spatial and feature dimensions. We found that cost volume normalization improves convergence and final performance in unsupervised optical flow. These findings are consistent with prior work that used a similar form of normalization to improve geometric matching [23].

**Unsupervised Learning Objectives:** Defining a learning objective  $\mathcal{L}(\cdot)$  that specifies the task of learning optical flow without having access to labels is the core problem of unsupervised optical flow. Similar to related work [14,15,18,30], we train our model by estimating optical flow and applying the respective losses in both directions. In this work we consider a learning objective that consists of three terms: occlusion-aware photometric consistency, edge-aware smoothness, and self-supervision, which we will now discuss in detail.

*Photometric Consistency:* The photometric consistency term encourages the estimated flow to align image patches with a similar appearance by penalizing photometric dissimilarity. The metric for measuring appearance similarity is critical for any unsupervised optical flow technique. Related approaches use three different objectives here (sometimes in combination), (i) the generalized Charbonnier loss [12,18,22,30,34,35], (ii) the structural similarity index (SSIM) loss [33,21,29], and (iii) the Census loss [18,35,36]. We compare all three losses in this paper. The generalized Charbonnier loss [24] is  $\mathcal{L}_C = \frac{1}{n} \sum ((I^{(1)} - w(I^{(2)})^2 + \epsilon^2)^\alpha$ . Our experiments use  $\epsilon = 0.001$  and  $\alpha = 0.5$  and also compare to using a modified L1 loss  $\mathcal{L}_{L1} = \sum |I^{(1)} - w(I^{(2)}) + \epsilon'|$  with  $\epsilon' = 10^{-6}$ . For the SSIM [31] loss, we use an occlusion-aware implementation from recent work [9]. For the Census loss, we

use a soft Hamming distance on Census-transformed image patches [18]. Based on the empirical results discussed below, we use the Census loss unless otherwise stated. All photometric losses are computed using an occlusion-masked average over all pixels [30].

*Occlusion Estimation:* By definition, occluded regions do not have a correspondence in the other image, so they should be discounted when computing the photometric loss. Related approaches estimate occlusions by (i) checking for consistent forward and backward flow [30], (ii) using the range map of the backward flow [3], and (iii) learning a model for occlusion estimation [12]. We are considering and comparing the first two variants here and improve the second variant through gradient stopping. In addition to taking into account occlusions, we also mask “invalid” pixels whose flow vectors point outside of the frame of the image [30]. The forward-backward consistency check defines occlusions a pixels for which the flow and the back-projected backward flow disagree by more than a threshold, such that the occlusion mask is defined as  $O^{(1)} = \mathbb{1}_{|V^{(1)} - w(V^{(2)})|^2 < \alpha_1 (|V^{(1)}|^2 - |w(V^{(2)})|^2) + \alpha_2}$ , where  $\alpha_1 = 0.01$  and  $\alpha_2 = 0.5$  [26]. An alternative approach computes a “range map”  $R^{(i)} \in \mathbb{R}^{H \times W}$  – a soft histogram of how many pixels in the other image map onto a given pixel, which is constructed by having each flow vector distribute a total weight of 1 to the four pixels around its end point according to a bilinear kernel [30]. Pixels that none of the reverse flow vectors point to are assumed to have no correspondence in the other image, and are therefore occluded. As proposed by Wang *et al.* [30], we compute an occlusion mask  $O^{(i)} \in \mathbb{R}^{W \times H}$  by thresholding the range map at 1. Based on the empirical results below, we use range-map based occlusion estimation by default, but use the forward-backward consistency check on KITTI, where it significantly improves performance.

*Gradient Stopping at Occlusion Masks:* Although prior work does not mention this issue [30], we found that propagating the gradient of the photometric loss into the occlusion estimation consistently degraded performance or caused divergence when the occlusion estimation was differentiable, as is the case for range-map based occlusion. This behavior is to be expected because when computing the occlusion-weighted average over photometric dissimilarity, there should be a gradient towards masking pixels with high photometric error. We address this problem by stopping the gradient at the occlusion mask, which eliminates divergence and improves performance.

*Smoothness:* Different forms of smoothness are commonly used to regularize optical flow in traditional methods [3,10,24] as well as most recent unsupervised approaches [34,22,30,18,33,36,21,12,29,35]. In this work, we consider edge-aware first and second order smoothness [27], where flows are encouraged to align their boundaries with visual edges in the image  $I^{(1)}$ . Formally, we define  $k$ th order smoothness as:

$$\mathcal{L}_{smooth(k)} = \frac{1}{n} \sum \exp\left(-\frac{\lambda}{3} \sum_c \left|\frac{\partial I_c^{(1,\ell)}}{\partial x}\right|\right) \left|\frac{\partial^k V^{(1,\ell)}}{\partial x^k}\right| + \exp\left(-\frac{\lambda}{3} \sum_c \left|\frac{\partial I_c^{(1,\ell)}}{\partial y}\right|\right) \left|\frac{\partial^k V^{(1,\ell)}}{\partial y^k}\right|. \quad (2)$$

Where  $\lambda$  modulates edge weighting based on  $I_c^{(1,\ell)}$  for color channel  $c \in [0, 2]$ . By default, we use first order smoothness on Flying Chairs and Sintel and second order smoothness on KITTI, which we ablate in different experiments.

*Smoothness at Flow Resolution:* A question that we have not seen addressed is at which level  $\ell$ , smoothness should be applied. Since we follow the commonly used method of estimating optical flow at  $\ell = 2$ , i.e. at a quarter of the input resolution, followed by upsampling through bilinear interpolation, our model produces piece-wise linear flow fields. As a result, only every fourth pixel can possibly have a non-zero second order derivative, which might not be aligned with the corresponding image edge and thereby reduce the effectiveness of edge-aware smoothness. To address this, we apply smoothness at level  $\ell = 2$  where flow is generated and downsample the image instead of upsampling the flow. This of course does not affect evaluation, which is done at the original image resolution.

*Self-supervision:* The idea of self-supervision in unsupervised optical flow is to generate optical flow labels by applying the learned model on a pair of images, then modify the images to make flow estimation more difficult and train the model to recover the originally estimated flow [14,15]. Since we see the main utility of this technique in learning flow estimation for pixels that go out of the image boundary – where cost-volume computation is not informative and photometric losses do not apply – we build on and improve ideas about self-supervised image crops [14]. For our self-supervised objective, we apply our model on the full images, crop the images by removing 64 pixels from each edge, apply the model again, and use the cropped estimated flow from the full images as supervision for flow estimation from the cropped images. We define the self-supervision objectives as an occlusion-weighted Charbonnier loss, that takes into account only pixels that have low forward-backward consistency in the “student” flow from cropped image and high forward-backward consistency in the “teacher” flow from the original images, similar to DDFlow [14].

*Continual Self-supervision and Image Resizing:* Unlike related work, we do not first train and then freeze a teacher model to supervise a separate student model but rather have a single model that supervises itself, which simplifies the approach, reduces the required memory, and allows the self-supervision signal to improve continually. To stabilize learning, we stop gradients of the self-supervision objectives to be propagated into the “teacher” flow. Additionally, we resize the image crops to match the original resolution before feeding them into the model (and we rescale the self-generated flow labels accordingly) to make the self-supervision examples more representative of the problem of extrapolating flow beyond the image boundary in the original size.

**Optimization:** To train our model  $f_\theta(\cdot)$  we minimize a weighted sum of losses:

$$\mathcal{L}(D, \theta) = w_{photo} \cdot \mathcal{L}_{photo} + w_{smooth} \cdot \mathcal{L}_{smooth} + w_{self} \cdot \mathcal{L}_{self}, \quad (3)$$

where  $\mathcal{L}_{photo}$  is our photometric loss,  $\mathcal{L}_{smooth}$  is smoothness regularization, and  $\mathcal{L}_{self}$  is the self-supervision Charbonnier loss. We set  $w_{photo}$  to 1 for experiments using the Census loss and to 2 when we compare to the SSIM, Charbonnier, or L1 losses. We set  $w_{self}$  to 2 when using first order, and to 4 for second order smoothness and use an edge-weight of  $\lambda = 150$ . We use  $w_{self} = 0$  during the first

half of training, linearly increase it 0.3 during the next 10% of gradient steps and keep it constant afterwards.

RGB image values are scaled to  $[-1, 1]$ , and augmented by randomly swapping the color channels and randomly shifting the hue. Sintel images are additionally randomly flipped up/down and left/right. All models are trained using Adam [13] ( $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\epsilon = 10^{-8}$ ) with a learning rate of  $10^{-4}$  for  $m$  steps, followed by another  $\frac{1}{5}m$  steps during which the learning rate is exponentially decayed to  $10^{-8}$ . All ablations use  $m = 50K$  with batch size 32, but the final model was trained using  $m = 1M$  with batch size 1, which produced slightly better performance as described below. Either way, the training takes about three days. Experiments on Sintel and KITTI start from a model that was first trained on Flying Chairs.

## 5 Experiments

We evaluate our model on the standard optical flow benchmark datasets: Flying Chairs [5], Sintel [4], and KITTI 2012/2015 [6,19]. We divide Flying Chairs and Sintel according to its standard train/test split. For KITTI, we train on the multi-view extension on the KITTI 2015 dataset, and we do not train on any data from KITTI 2012 because it does not have moving objects.

Related work is inconsistent in their use of train/test splits. For Sintel, it is common to train on the training set, report the benchmark performance on the test set, and evaluate ablations on the training set only (because test set labels are not public), which does not test generalization very well. Others “download the Sintel movie and extract ~10,000 images” [15] including the test set images, which is intended to demonstrate the ability of unsupervised methods to train on raw video data, but unfortunately also includes the benchmark test images in the training set. For KITTI, other works train on the raw KITTI dataset with and without excluding the evaluation set, or most commonly train on frames 1–8 and 13–20 of the multi-view extension of KITTI 2012/2015 datasets and evaluate on frames 10/11. But this split can mask overfitting to the trained sequences – either in the ablation results or also in the benchmark results, when the multiview-extensions of both the train and the test set are used. We therefore adopt the training regimen of Zhong *et al.* [35] and train two models for each dataset, one on the training set and one on test set (or for KITTI on their multiview extension) and evaluate these models appropriately.

Following the conventions of the KITTI benchmark, we report endpoint error (“EPE”) and error rates (“ER”), where a prediction is considered erroneous if its EPE is  $> 3$  pixels and if the distance between the predicted point and the true end point is  $> 5\%$  of the length of the true flow vector. We compute these metrics for all pixels (“occ” in the KITTI benchmark, which we call “all” in this paper). We use the common practice of pretraining on the train split of the Flying Chairs dataset before training on Sintel / KITTI. We evaluate on all images in the native resolution, but have the model perform inference on a resolution that is divisible by 32, output at a four times smaller resolution,

and then resize the output to the original resolution for evaluation. On KITTI, we observe that performance improves when using a square input resolution instead of a resolution in the original aspect ratio – perhaps because KITTI is dominated by horizontal motion. Accordingly, we use the following resolutions in our experiments: Flying Chairs:  $384 \times 512$ , Sintel:  $448 \times 1024$ , KITTI:  $640 \times 640$ .

## 6 Results

We evaluate our model in an extensive comparison and ablation study, from which we identify the best combination of components, tested in the “full” setting, which is often different from the components that work best individually in our “minimal” setting (more details below). We then compare our resulting model to the best published methods on unsupervised optical flow, and show that it outperforms all methods on all benchmarks.

**Ablations and Comparisons of Key Components** To determine which aspects of unsupervised optical flow are most important, we perform an extensive series of ablation studies. We find that a) occlusion-masking, self-supervision, and smoothness are all important, b) level dropout and cost volume normalization improve performance, c) the Census loss outperforms other photometric losses, d) range-map based occlusion estimation requires gradient stopping to work, e) edge-aware smoothness and smoothness level matters significantly, f) self supervision helps especially for KITTI, and is improved by our changes, g) losses might be the current performance bottleneck, f) changing the resolution can substantially improve results, g) data augmentation and pretraining are helpful.

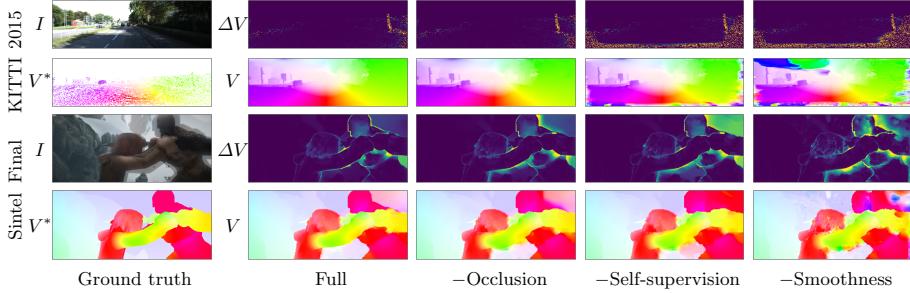
In each ablation study we train one model per domain (on Flying Chairs, KITTI-test, and Sintel-test), and evaluate those on the corresponding validation split from the same domain, taking into account occluded and non-occluded pixels “(all)”. To estimate the noise in our results, we trained models with six different random seeds for each domain and computed their standard deviations per metric: Flying Chairs: 0.0162, Sintel Clean: 0.0248, Sintel Final: 0.0131, KITTI-2015: 0.0704, 0.0718%. We now describe the findings of each study.

*Core Components:* Table 1 shows how performance varies as each core component of our model (occlusion masking, smoothness, and self-supervision) is removed. We see that every component contributes to the overall performance. Since the utility of different components depends on what other components are used, all following experiments compare to the “minimal” (first row) and “full” (last row) versions of our method. Qualitative results for rows

9, 4, 3, and 1 are shown in Figure 2 (from left to right). Note how the flow error  $\Delta V$  increases with each removal of a core component.

**Table 1.** Core components: OM: occlusion masking, SM: smoothness, SS: self-supervision; “div.”: divergence

OM	SM	SS	Chairs			Sintel train			KITTI-15 train		
			test	Clean	Final	all	noc	ER%	all	noc	ER%
–	–	–	3.58	4.20	6.80	13.07	2.47	21.21			
–	–	✓	2.99	3.34	5.18	11.36	2.30	18.61			
–	✓	–	2.84	3.37	5.19	11.37	2.17	19.31			
–	✓	✓	2.74	3.12	4.56	3.28	2.08	9.97			
✓	–	–	3.28	3.78	5.85	div.	div.	div.			
✓	–	✓	2.91	3.26	4.72	3.02	2.11	9.89			
✓	✓	–	2.63	3.20	4.63	4.15	2.05	13.15			
✓	✓	✓	2.55	3.00	4.18	2.94	1.98	9.65			



**Fig. 2.** Qualitative ablation results of our model on random images not seen during training. Flow quality deteriorates as we progressively ablate core components

*Model Improvements:* Table 2 shows that level dropout (LD) and cost volume normalization (CVN) improve performance in the full setting (but not generally in the minimal setting). CVN appears to be more important for Chairs and Sintel while LD helps most for KITTI.

*Photometric Losses:* Table 3 compares commonly used photometric losses and shows that it is important to test every component with the full method, rather than looking at isolated performance. By itself, the commonly-used Charbonnier loss works better, but in the full setting, it underperforms the simpler L1 loss. For KITTI, Census works best in both settings. But for Sintel (in particular Sintel Final), the SSIM loss significantly outperforms Census in the minimal setting (5.41 vs. 6.98) but does not perform as well when used with all components in the full setting.

*Occlusion Estimation:* Table 4 compares different approaches to occlusion estimation (forward-backward consistency and range maps). We see that range-map based occlusion consistently diverges unless we stop the gradient of the photometric loss. But when gradients are stopped, this method works well, especially for Flying Chairs and Sintel Clean. Forward-backward consistency works best for KITTI, especially if not applied from the beginning.

**Table 2.** Model improvements. CVN: cost volume normalization, LD: level dropout

	CVN	LD	Chairs		Sintel train		KITTI-15 train		
			test		Clean	Final	all	noc	ER%
Minimal	–	–	5.01	4.52	6.67	13.30	2.72	21.69	
	–	✓	5.29	4.40	<b>6.59</b>	<b>12.75</b>	2.49	21.30	
	✓	–	4.86	<b>4.19</b>	6.69	13.294	2.59	21.54	
Full	✓	✓	<b>3.58</b>	4.20	6.80	13.07	<b>2.47</b>	<b>21.21</b>	
	–	–	3.78	3.41	4.70	39.09	30.19	98.77	
	–	✓	3.21	3.45	4.61	2.96	<b>1.96</b>	9.77	
	✓	–	<b>2.54</b>	3.07	4.31	3.16	2.04	10.35	
	✓	✓	2.55	<b>3.00</b>	<b>4.18</b>	<b>2.94</b>	1.98	<b>9.65</b>	

**Table 3.** Photometric losses. Best results of L1 and Charbonnier underlined

Method	Chairs		Sintel train		KITTI-15 train		
	test		Clean	Final	all	noc	ER%
Minimal	L1	<u>4.27</u>	5.51	7.74	17.02	6.11	32.96
	Charbonnier	4.31	<u>5.50</u>	<u>7.64</u>	<u>16.94</u>	<u>6.09</u>	<u>32.84</u>
	SSIM	<u>3.51</u>	<u>4.01</u>	<u>5.41</u>	11.99	2.46	21.72
	Census	3.54	4.23	6.98	<b>11.66</b>	<b>2.37</b>	<b>21.15</b>
Full	L1	<u>2.83</u>	<u>4.23</u>	<u>5.75</u>	<u>5.53</u>	<u>3.17</u>	<u>18.65</u>
	Charbonnier	2.86	4.24	5.81	5.56	3.21	18.82
	SSIM	<u>2.54</u>	3.08	4.52	3.29	2.04	10.41
	Census	2.61	<u>3.00</u>	<u>4.20</u>	<u>3.08</u>	<u>2.01</u>	<u>10.01</u>

**Table 4.** Occlusion estimation. RM: range-map based occlusion, FB: forward-backward consistency check

Method	Chairs		Sintel train		KITTI-15 train		
	test		Clean	Final	all	noc	ER%
Minimal	None	3.51	4.15	6.69	12.89	2.41	21.17
	RM (w/o grad stop)	div.	div.	div.	div.	div.	div.
	RM (w/ grad stop)	<b>3.27</b>	3.78	5.86	10.65	2.29	18.76
	FB (from step 1)	3.57	<b>3.71</b>	<b>4.83</b>	<b>8.99</b>	2.16	<b>17.71</b>
	FB (after 20% steps)	3.49	3.76	4.92	9.75	<b>2.13</b>	18.38
	None	2.73	3.84	5.13	3.28	2.10	10.07
Full	RM (w/o grad stop)	div.	div.	div.	div.	div.	div.
	RM (w/ grad stop)	<b>2.58</b>	<b>3.01</b>	4.25	3.10	2.04	9.86
	FB (from step 1)	3.28	3.49	4.45	2.96	1.99	9.65
	FB (after 20% steps)	3.14	3.12	<b>4.13</b>	<b>2.88</b>	<b>1.95</b>	<b>9.54</b>

*Smoothness:* Prior work suggests that photometric and smoothness losses taken together work better at higher resolutions [8]. But our analysis of the smoothness loss alone shows an advantage of applying this loss at the resolution of flow estimation, rather than at the image resolution, in particular for Flying Chairs and Sintel (Table 5). Our results also show that first order smoothness works better on Chairs and Sintel while second order smoothness works better on KITTI (Table 6). We see that context is important because in the minimal setting, the best second order smoothness weight for KITTI is 8, but in the full setting, it is 2. Comparing different edge-weights  $\lambda$  (Eq. 2) in Table 7, we see that nonzero edge-weights improve performance, particularly in the full setting. To our surprise, the simple strategy of only optimizing the Census loss and second order smoothness without edge-awareness, occlusion, or self-supervision (first row) produces performance on KITTI that improves on previous the state of the art.

*Self-Supervision:* In Table 8 we ablate the use of self-supervision and our proposed changes, and confirm that self-supervision on image crops is instrumental in achieving good results on KITTI, where errors are dominated by fast motion near the image edges. We also see that self-supervision is most effective when the image crop is resized as proposed by our method. Freezing the teacher network, as done in other works, seems to be important only when not using the other regularizing components. With these components in place, sharing the same model for both student and teacher appears to be beneficial.

*Loss Comparison to Ground Truth:* Photometric loss functions used in unsupervised optical flow rely on the brightness consistency assumption: that pixel intensities in the camera image are invariant to motion in the world. But pho-

**Table 5.** Level for smoothness loss

Smoothn. level	Chairs <i>test</i>	Sintel train		KITTI-15 train		
		Clean	Final	all	noc	ER%
Minimal	0	3.05	4.10	5.22	12.16	2.32
	1	2.94	3.65	<b>5.07</b>	11.94	2.24
	2	<b>2.85</b>	<b>3.33</b>	5.21	<b>11.43</b>	<b>2.23</b>
Full	0	2.87	3.65	4.63	2.95	2.02
	1	2.74	3.13	4.29	2.96	<b>1.99</b>
	2	<b>2.58</b>	<b>3.00</b>	<b>4.24</b>	<b>2.93</b>	<b>1.99</b>
						<b>9.63</b>

**Table 6.** Comparison of weights for first/second order smoothness

<i>w</i> <sub>smooth</sub> 1st 2nd	Chairs <i>test</i>	Sintel train		KITTI-15 train		
		Clean	Final	all	occ	ER%
Minimal	0 0	4.55	4.16	6.84	div.	div.
	0 2	3.13	3.77	6.32	11.37	2.17
	0 8	4.02	3.50	6.08	<b>7.27</b>	<b>2.11</b>
Full	4 0	<b>2.85</b>	<b>3.35</b>	<b>5.05</b>	7.23	2.30
	16 0	4.37	4.78	6.03	9.58	4.09
	0 0	2.92	3.27	4.77	2.92	2.07
Minimal	0 2	2.79	div.	div.	<b>2.93</b>	<b>1.98</b>
	0 8	2.75	3.33	4.77	2.94	<b>1.91</b>
	4 0	<b>2.60</b>	<b>3.00</b>	<b>4.17</b>	5.39	2.03
Full	16 0	3.68	4.22	5.30	8.71	4.01
						21.52

**Table 7.** Smoothness edge-weights

$\lambda$	Chairs <i>test</i>	Sintel train		KITTI-15 train		
		Clean	Final	all	noc	ER%
Minimal	0	4.93	6.00	6.65	<b>4.15</b>	2.36
	10	4.33	5.32	6.12	4.22	<b>2.17</b>
	150	<b>2.83</b>	<b>3.36</b>	<b>5.12</b>	11.41	2.21
Full	0	4.87	5.78	6.40	3.86	2.84
	10	3.75	4.62	5.34	3.14	2.11
	150	<b>2.56</b>	<b>3.02</b>	<b>4.20</b>	<b>2.87</b>	<b>1.95</b>
						<b>9.59</b>

**Table 8.** Self-supervision ablation

Self-supervision	Chairs <i>test</i>	Sintel train		KITTI-15 train		
		Clean	Final	all	noc	ER%
Minimal	None	3.48	4.10	6.62	13.05	2.48
	No resize	3.16	3.53	5.67	12.87	2.35
	Frozen teacher	3.10	3.36	5.24	<b>8.11</b>	<b>2.38</b>
Full	Default	<b>2.99</b>	<b>3.34</b>	<b>5.18</b>	11.36	2.30
	None	2.67	3.18	4.60	4.10	2.02
	No resize	<b>2.51</b>	3.14	4.48	3.53	2.02
Full	Frozen teacher	2.66	3.04	4.24	2.99	1.99
	Default	2.61	<b>2.99</b>	<b>4.23</b>	<b>2.86</b>	<b>1.95</b>
						<b>9.57</b>

**Table 9.** Losses on Sintel for zero flow, ground truth flow, and predicted flow

Flow	L1		SSIM		Census		SM		Census + SM	
	noc	all	noc	all	noc	all	all	noc	all	
Clean	Zero	.146	.161	.927	.946	3.160	3.193	0	3.160	3.193
	GT	<b>.031</b>	.052	<b>.191</b>	<b>.241</b>	<b>2.041</b>	<b>2.122</b>	.032	<b>2.073</b>	<b>2.154</b>
	UFlow	<b>.031</b>	<b>.042</b>	.203	.247	2.06	2.130	<b>.024</b>	2.085	<b>2.154</b>
Final	Zero	.126	.142	.731	.751	3.037	3.075	0	3.037	3.075
	GT	.034	.055	.185	.233	2.086	2.154	.063	2.149	2.217
	UFlow	<b>.032</b>	<b>.037</b>	.167	.226	<b>2.044</b>	<b>2.091</b>	.045	<b>2.089</b>	<b>2.136</b>

tometric consistency is an imperfect indicator of flow quality (e.g. in regions of shadows and specularity). To analyze this issue, we compute photometric and smoothness losses not only for the flow field produced by our model, but also for a flow field filled with zeros and for the ground truth flow. Table 9 shows that our model is able to achieve comparable or better photometric consistency (and overall loss) than the ground truth flow. This trend is more pronounced on Sintel Final, which we believe violates the consistency assumption more than Sintel Clean. This result suggests that the loss functions currently used may be a limiting factor in unsupervised methods.

*Resolution:* Table 10 shows, perhaps surprisingly, that estimating flow at a different resolution and aspect ratio can substantially improve performance on KITTI-15 (2.93 vs. 3.80), presumably because the motion field in this dataset is dominated by horizontal motion. We have not observed this effect in other datasets.

*Data Augmentation:* Table 11 evaluates the importance of color augmentation (color channel swapping and hue randomization) for all domains, as well as image flipping for Sintel. The results show that both augmentation techniques improve performance, in particular image flipping for Sintel (which is a much smaller dataset than Chairs or KITTI).

*Pretraining:* Pretraining is a common strategy in supervised [5,25] and unsupervised [14,35] optical flow. The results in Table 12 confirm that pretraining on Chairs improves performance on Sintel and KITTI.

*Gradient Steps and Batch Size:* All experiments up to this point have trained the model for 60K steps at a batch size of 32. Table 13 shows a comparison to another training regime that trains longer with smaller batches, which consistently improves performance. We use this regime for our comparison to other published methods.

**Comparison to State of the Art:** We show qualitative results in Figure 3 and quantitatively evaluate our model trained on KITTI and Sintel data in the corresponding benchmarks in Table 14, where we compare against state-of-the-art techniques for unsupervised and supervised optical flow. Results not reported by prior work are indicated with “–”.

**Table 10. Resolution**

Resolution	KITTI-15 train			
	all	noc	ER%	
Min.	384×1280	13.25	2.79	21.38
	640×640	12.91	2.42	21.17
Full	384×1280	3.80	2.13	10.88
	640×640	<b>2.93</b>	<b>1.96</b>	<b>9.61</b>

**Table 11. Data augmentation.** F: image flipping up/down and left/right (not used for KITTI), C: color augmentation

F	C	Chairs		Sintel train		KITTI-15 train			
		<i>test</i>		Clean	Final	all	noc	ER%	
		–	–	<b>3.47</b>	4.39	<b>6.56</b>	13.27	2.56	22.13
Minimal	–	✓	–	3.56	4.38	6.58	<b>13.07</b>	<b>2.47</b>	<b>21.21</b>
	✓	–	–	3.49	4.23	6.73	–	–	–
	✓	✓	–	3.58	<b>4.20</b>	6.80	–	–	–
	–	–	–	2.53	3.84	5.14	3.06	2.03	9.82
Full	–	✓	–	2.61	3.78	5.23	<b>2.94</b>	<b>1.98</b>	<b>9.65</b>
	✓	–	–	2.57	3.02	4.22	–	–	–
	✓	✓	–	<b>2.55</b>	<b>3.00</b>	<b>4.18</b>	–	–	–

**Table 12. Pretraining on Chairs**

Pretraining on Chairs	Sintel train		KITTI-15 train			
	Clean	Final	all	noc	ER%	
Min.	–	4.41	7.53	<b>12.93</b>	<b>2.44</b>	21.24
	✓	<b>4.20</b>	<b>6.80</b>	13.07	2.47	<b>21.21</b>
Full	–	3.38	4.81	3.08	2.04	10.00
	✓	<b>3.00</b>	<b>4.18</b>	<b>2.94</b>	<b>1.98</b>	<b>9.65</b>

**Table 13. Gradient steps (S) and batch size (B)**

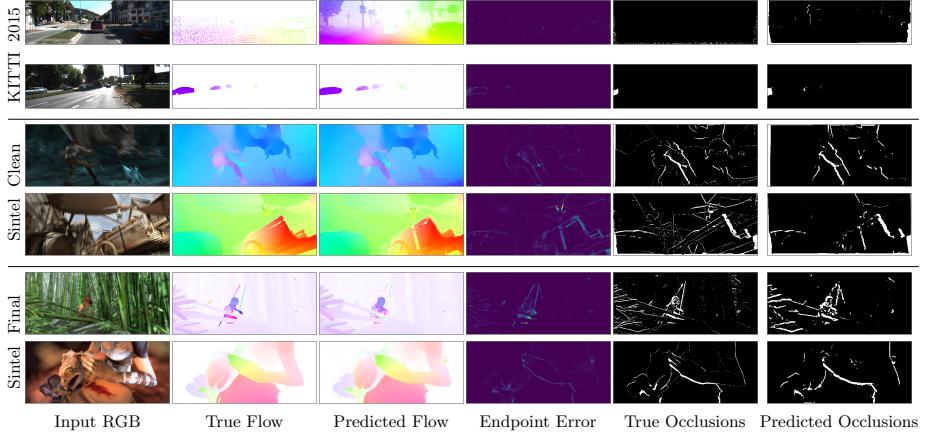
S	B	Chairs		Sintel train		KITTI-15 train		
		<i>test</i>	Clean	Final	all	noc	ER%	
train	60K	32	{3.16}	3.04	4.23	2.92	<b>1.96</b>	9.71
	1.2M	1	{2.82}	<b>3.01</b>	<b>4.09</b>	<b>2.84</b>	<b>1.96</b>	<b>9.39</b>
	60K	32	2.57	{2.47}	{3.92}	{2.74}	{1.87}	{9.04}
	1.2M	1	<b>2.55</b>	{2.50}	{3.39}	{2.71}	{1.88}	{9.05}

**Table 14.** Our model (yellow) compared to state of the art. Supervised models in gray fine-tune on their evaluation domain, which is often not possible in practice. Braces indicate models whose training set includes its evaluation set, and so are not comparable: “()” trained on the labeled evaluation set, “{}” trained on the unlabeled evaluation set, and “[ ]” trained on data related to the evaluation set (e.g. < 5 frames away in KITTI, or having the same content in Sintel). The best unsupervised and supervised (without finetuning) results are in bold. Methods that use additional modalities are denoted with MDM: mono depth/motion, SDM: stereo depth/motion, MF: multi-frame flow

Method	Sintel Clean [4]				Sintel Final [4]				KITTI 2012 [6]				KITTI 2015 [19]			
	EPE		EPE		EPE		EPE		EPE		EPE (noc)		ER in %			
	train	test	train	test	train	test	train	test	train	train	train	test	train	test	train	test
Supervised	(A) FlowNet2-ft [11]	(1.45)	4.16	(2.01)	5.74	(1.28)	1.8	(2.30)	—	—	(8.61)	11.48	—	—	—	—
	(B) PWC-Net-ft [25]	(1.70)	3.86	(2.21)	5.13	(1.45)	1.7	(2.16)	—	—	(9.80)	9.60	—	—	—	—
	(C) SelFlow-ft [15]	(1.68)	[3.74]	(1.77)	{4.26}	(0.76)	1.5	(1.18)	—	—	—	8.42	—	—	—	—
	(D) VCN-ft [32]	(1.66)	2.81	(2.24)	4.40	—	—	(1.16)	—	—	(4.10)	6.30	—	—	—	—
	(E) FlowNet2 [11]	<b>2.02</b>	<b>3.96</b>	<b>3.14</b>	<b>6.02</b>	<b>4.09</b>	—	9.84	—	—	28.20	—	—	—	—	—
	(F) PWC-Net [25]	2.55	—	3.93	—	4.14	—	10.35	—	—	33.67	—	—	—	—	—
	(G) VCN [32]	2.21	—	3.62	—	—	—	<b>8.36</b>	—	—	<b>25.10</b>	—	—	—	—	—
Unsupervised	(H) Back2Basics [34]	—	—	—	—	11.30	9.9	—	—	—	—	—	—	—	—	—
	(I) DSTFlow [22]	{6.16}	10.41	{7.38}	11.28	[10.43]	12.4	[16.79]	[6.96]	[36.00]	[39.00]	—	—	—	—	—
	(J) OAFlow [30]	{4.03}	7.95	{5.95}	9.15	[3.55]	[4.2]	[8.88]	—	—	[31.20]	—	—	—	—	—
	(K) UnFlow [18]	—	—	7.91	10.21	3.29	—	8.10	—	—	23.27	—	—	—	—	—
	(L) GeoNet [33] (MDM)	—	—	—	—	—	—	10.81	8.05	—	—	—	—	—	—	—
	(M) DF-Net [36] (MDM)	—	—	—	—	3.54	4.4	{8.98}	—	{26.01}	{25.70}	—	—	—	—	—
	(N) CCFflow [21] (MDM)	—	—	—	—	—	—	5.66	—	20.93	25.27	—	—	—	—	—
	(O) MFOccFlow [12] (MF)	{3.89}	7.23	{5.52}	8.81	—	—	[6.59]	[3.22]	—	22.94	—	—	—	—	—
	(P) UnOS [29] (SDM)	—	—	—	—	1.64	<b>1.8</b>	5.58	—	—	18.00	—	—	—	—	—
	(Q) EPIFlow [35]	3.94	7.00	5.08	8.51	2.61	3.4	5.56	2.56	—	16.95	—	—	—	—	—
	(R) DDFlow [14]	{2.92}	6.18	{3.98}	7.40	[2.35]	3.0	[5.72]	[2.73]	—	14.29	—	—	—	—	—
	(S) SelFlow [15] (MF)	[2.88]	[6.56]	{3.87}	{6.57}	[1.69]	2.2	[4.84]	[2.40]	—	14.19	—	—	—	—	—
	(T) UFlow-test	<b>3.01</b>	—	<b>4.09</b>	—	<b>1.58</b>	—	<b>2.84</b>	<b>1.96</b>	<b>9.39</b>	—	—	—	—	—	—
	(U) UFlow-train	{2.50}	<b>5.21</b>	{3.39}	<b>6.50</b>	1.68	1.9	{2.71}	{1.88}	{9.05}	<b>11.13</b>	—	—	—	—	—

Among unsupervised approaches (H-U), our model sets a new state of the art for Sintel Clean (5.21 vs. 6.18), Sintel Final (6.50 vs. 7.40), and KITTI-15 (11.13% vs. 14.19%) – where, for a lack of comparability, we had to disregard results in braces that came from (partially) training on the test set. UFlow is only outperformed (1.8 vs. 1.9) on KITTI-12, which does not include moving objects, by a stereo-depth and motion based approach (P).

The top-performing supervised models finetuned on data from the evaluation domain (models A-D) do outperform our unsupervised model, as one may expect. But on KITTI-15, our model *performs on par with the supervised FlowNet2*. Of course, fine-tuning on the domain is only possible because the KITTI training data also contains ground-truth flow, which we ignore but which supervised techniques require. This sort of supervision is hard to obtain (KITTI being virtually the only non-synthetic dataset with this information), which demonstrates the value of unsupervised flow techniques such as ours. Without access to the ground truth labels of the test domain, our unsupervised method compares more favorably to its supervised counterparts, significantly outperforming them on KITTI. Our final experiment analyses cross-domain generalization in more detail.



**Fig. 3.** Results for our model on random examples not seen during training taken from KITTI 2015 and Sintel Final. These qualitative results show the model’s ability to estimate fast motions, relatively fine details, and substantial occlusions

Table 15 evaluates out-of-domain generalization by training and evaluating models across three datasets. While performance is best when training and test data are from the same domain, our model shows good generalization. It consistently outperforms DDFlow and it outperforms the supervised PWC-Net in all but one generalization task (training on Chairs and testing on Sintel Clean).

**Table 15.** Generalization across datasets. Performance when training on one dataset and testing on different one (gray if same)

Method	Chairs	Sintel train		KITTI-15 train			
	test	Clear	Final	all	noc	ER%	
Train on Chairs	PWC-Net [25]	2.00	<b>3.33</b>	4.59	13.20	—	41.79
	DDFlow [14]	2.97	4.83	4.85	17.26	—	—
	UFlow-test	{2.82}	4.36	5.12	15.68	7.96	32.69
	UFlow-train	2.55	3.43	<b>4.17</b>	<b>11.27</b>	<b>5.66</b>	<b>30.31</b>
Train on Sintel	PWC-Net [25]	3.69	(1.86)	(2.31)	10.52	—	30.49
	DDFlow [14]	3.46	{2.92}	{3.98}	12.69	—	—
	UFlow-test	3.39	3.01	4.09	<b>7.67</b>	<b>3.77</b>	<b>17.41</b>
	UFlow-train	<b>3.25</b>	(2.50)	{3.39}	9.40	4.53	20.02
Train on KITTI	PWC-Net [25]	6.35	6.20	7.08	[5.72]	—	—
	DDFlow [14]	5.25	6.34	7.01	2.84	1.96	9.39
	UFlow-test	5.05	<b>5.58</b>	<b>6.31</b>	{2.71}	{1.88}	{9.05}
	UFlow-train	—	—	—	—	—	—

## 7 Conclusion

We have presented a study into what matters in unsupervised optical flow that systematically analyzes, compares, and improves a set of key components. This study results in a range of novel observations about these components and their interactions, from which we integrate the best components and improvements into a unified framework for unsupervised optical flow. Our resulting UFlow model substantially outperforms the state of the art among unsupervised methods and performs on par with the supervised FlowNet2 on the challenging KITTI 2015 benchmark, despite not using any labels. In addition to its strong performance, our method is also significantly simpler than many related approaches, which we hope will make it useful as a starting point for further research into unsupervised optical flow. Our code is available at <https://github.com/google-research/google-research/tree/master/uflow>.

## References

1. Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 2011.
2. John L. Barron, David J. Fleet, and Steven S. Beauchemin. Performance of optical flow techniques. *IJCV*, 1994.
3. Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. *ECCV*, 2004.
4. Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. *ECCV*, 2012.
5. Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. *ICCV*, 2015.
6. Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. *CVPR*, 2012.
7. James J. Gibson. *The Perception of the Visual World*. Houghton Mifflin, 1950.
8. Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. *ICCV*, 2019.
9. Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. *ICCV*, 2019.
10. Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 1981.
11. Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. *CVPR*, 2017.
12. Joel Janai, Fatma Güney, Anurag Ranjan, Michael J. Black, and Andreas Geiger. Unsupervised learning of multi-frame optical flow with occlusions. *ECCV*, 2018.
13. Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.
14. Pengpeng Liu, Irwin King, Michael R. Lyu, and Jia Xu. DDFlow: Learning optical flow with unlabeled data distillation. *AAAI*, 2019.
15. Pengpeng Liu, Michael R. Lyu, Irwin King, and Jia Xu. Selfflow: Self-supervised learning of optical flow. *CVPR*, 2019.
16. Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. *DARPA Image Understanding Workshop*, 1981.
17. Nikolaus Mayer, Eddy Ilg, Philipp Fischer, Caner Hazirbas, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. What makes good synthetic training data for learning disparity and optical flow estimation? *IJCV*, 2018.
18. Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. *AAAI*, 2018.
19. Moritz Menze, Christian Heipke, and Andreas Geiger. Joint 3d estimation of vehicles and scene flow. *ISPRS Workshop on Image Sequence Analysis*, 2015.
20. Anurag Ranjan and Michael J. Black. Optical flow estimation using a spatial pyramid network. *CVPR*, 2017.
21. 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. *CVPR*, 2019.
22. Zhe Ren, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Yang, and Hongyuan Zha. Unsupervised deep learning for optical flow estimation. *AAAI*, 2017.

23. Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. *CVPR*, 2017.
24. Deqing Sun, Stefan Roth, and Michael J. Black. Secrets of optical flow estimation and their principles. *CVPR*, 2010.
25. Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. *CVPR*, 2018.
26. Narayanan Sundaram, Thomas Brox, and Kurt Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. *ECCV*, 2010.
27. Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. *ICCV*, 1998.
28. A. Torralba and A. A. Efros. Unbiased look at dataset bias. *CVPR*, 2011.
29. Yang Wang, Peng Wang, Zhenheng Yang, Chenxu Luo, Yi Yang, and Wei Xu. Unos: Unified unsupervised optical-flow and stereo-depth estimation by watching videos. *CVPR*, 2019.
30. Yang Wang, Yi Yang, Zhenheng Yang, Liang Zhao, Peng Wang, and Wei Xu. Occlusion aware unsupervised learning of optical flow. *CVPR*, 2018.
31. Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004.
32. Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. *NeurIPS*, 2019.
33. Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. *CVPR*, 2018.
34. Jason J. Yu, Adam W. Harley, and Konstantinos G. Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. *ECCV Workshop*, 2016.
35. Yiran Zhong, Pan Ji, Jianyuan Wang, Yuchao Dai, and Hongdong Li. Unsupervised deep epipolar flow for stationary or dynamic scenes. *CVPR*, 2019.
36. Yuliang Zou, Zelun Luo, and Jia-Bin Huang. DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency. *ECCV*, 2018.