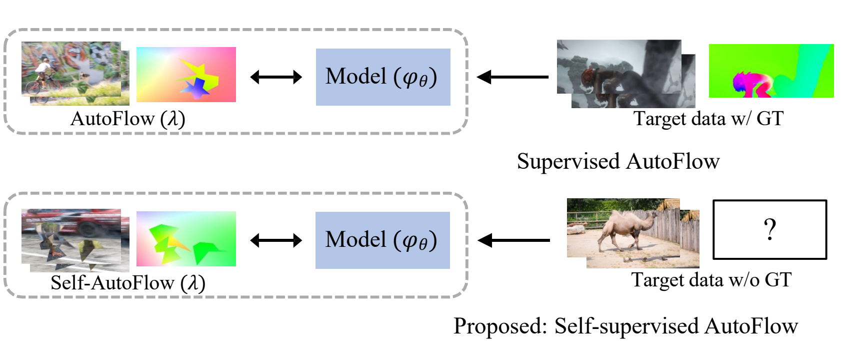
Self-supervised AutoFlow

Hsin-Ping Huang[1,2], Charles Herrmann[1], Junhwa Hur[1], Erika Lu[1],
Kyle Sargent[1], Austin Stone[1], Ming-Hsuan Yang[1,2], Deqing Sun[1]
[1]Google Research [2]University of California, Merced



	Sintel Final ↓	KITTI ↓	DAVIS ↑
Supervised AutoFlow	**2.41**	**3.86**	50.4
Proposed: Self-supervised AutoFlow	2.59	4.22	**51.9**

Figure 1. **Self-supervised AutoFlow** learns to generate an optical flow training set through self-supervision on the target domain. It performs comparable to supervised AutoFlow on Sintel and KITTI without requiring ground truth (GT) and learns a better dataset for real-world DAVIS, where GT is not available. We report optical flow accuracy on Sintel and KITTI, and keypoint propagation accuracy on DAVIS.

Abstract

Recently, AutoFlow has shown promising results on learning a training set for optical flow, but requires ground truth labels in the target domain to compute its search metric. Observing a strong correlation between the ground truth search metric and self-supervised losses, we introduce self-supervised AutoFlow to handle real-world videos without ground truth labels. Using self-supervised loss as the search metric, our self-supervised AutoFlow performs on par with AutoFlow on Sintel and KITTI where ground truth is available, and performs better on the real-world DAVIS dataset. We further explore using self-supervised AutoFlow in the (semi-)supervised setting and obtain competitive results against the state of the art.

1. Introduction

Data is the new oil. — Clive Humby, 2006 [13]

This well-known analogy not only foretold the critical role of data for developing AI algorithms in the last decade but also revealed the importance of *data curation*. Like refined oil, data must be carefully curated to be useful for AI algorithms to succeed. For example, one key ingredient for the success of AlexNet [21] is ImageNet [36], a large dataset created by extensive manual labeling.

The manual labeling process, however, is either not applicable or difficult to scale to many low-level vision tasks, such as optical flow. A common practice for optical flow is to pre-train models using large-scale synthetic datasets, *e.g.*, FlyingChairs [6] and FlyingThings3D [26], and then fine-tune them on limited in-domain datasets, *e.g.*, Sintel [4] or KITTI [28]. While this two-step process works better than directly training on the limited target datasets, there exists a domain gap between synthetic data and the target domain.

To narrow the domain gap, AutoFlow [41] learns to render a training dataset to optimize performance on a target dataset, obtaining superior results on Sintel and KITTI where the ground truth is available. As obtaining ground truth optical flow for most real-world data is still an open challenge, it is of great interest to remove this dependency on ground truth to apply AutoFlow to real-world videos.

In this paper, we introduce a way to remove this reliance by connecting learning to render with another independent line of research on optical flow, self-supervised learning (SSL). SSL methods for optical flow [15, 23–25, 53] use a set of self-supervised losses to train models using only image pairs in the target domain. We observe a strong correlation between these self-supervised losses and the ground truth errors, as shown in Fig. 2. This motivates us to connect these two lines of research by adopting self-supervised losses as a search metric for AutoFlow [41], calling our approach “Self-supervised AutoFlow”.

Self-supervised AutoFlow obtains similar performance to AutoFlow on Sintel [4] and KITTI [28], and it can learn a better dataset for the real-world DAVIS data [29] where ground truth is not available. To further narrow the domain gap between synthetic data and the target domain, we also explore new ways to better synergize techniques from learning to render and self-supervised learning.

Numerous self-supervised methods still rely on pre-training on a synthetic dataset. Our method replaces this pre-training with supervised training on self-supervised AutoFlow data generated using self-supervised metrics. This new pipeline is still self-supervised and obtains competitive performance among all self-supervised methods. We further demonstrate that our method provides a strong initialization for supervised fine-tuning and obtains competitive results against the state of the art.

We make the following main contributions:

- We introduce self-supervised AutoFlow to learn to render a training set for optical flow using self-supervision on the target domain, connecting two independently studied directions for optical flow: learning to render and self-supervised learning.
- Self-supervised AutoFlow performs competitively against AutoFlow [41] that uses ground truth on Sintel and KITTI and better on DAVIS where ground truth is not available.
- We further analyze self-supervised AutoFlow in semi-supervised and supervised settings and obtain competitive performance against the state of the art.

2. Related Work

CNN architectures for optical flow. Recent advances in deep learning and synthetic datasets have contributed to the development of numerous optical flow architectures. Early work introduces basic designs using U-Net [6, 14, 35] or an image pyramid [32]. PWC-Net [42], concurrently with LiteFlowNet [12], introduces an advanced design based on well-established domain knowledge (*e.g.*, pyramid, warping, and cost volume). RAFT [43] further advances architecture designs based on a full 4D cost volume with a recurrent optimizer, which significantly improves the accuracy and encourages many follow-up methods [18,19,40,45,52], followed by recent attention-based designs [11, 17, 39, 46] as well. As our main focus is on the dataset, we adopt the widely-used RAFT architecture in our experiments.

Self-supervised optical flow. Supervised approaches may not generalize well to real-world domains where annotations are difficult to obtain. To overcome the limitation, self-supervised approaches [1,33,49,53] directly train the networks on the target data with hand-crafted self-supervised losses [15,23–25,27,44]. UFlow [20] systematically analyzes the effect of various loss designs on the accuracy and proposes an optimized combination for the best accuracy. SMURF [38] presents a self-supervised method based on the RAFT [43] architecture and proposes several technical designs such as the sequence loss, full image warping, heavy augmentation, and multi-frame training. In this paper, we find that there is a strong correlation between self-supervised loss and ground truth errors, which inspires us to employ the self-supervised loss as a search metric for synthetic dataset learning. We further explore ways to synergize self-supervised methods and learning to render for better performance in the self-supervised setting.

Semi-supervised optical flow. To benefit from training on both labeled (out-of-domain) data and target domains, semi-supervised approaches propose to reduce a domain gap between datasets by using a GAN [7, 22], to learn a conditional prior from labeled data [48], to benefit from a small fraction of labels by active learning [50], or to adapt to the target domain through knowledge distillation [16]. Semi-Flow [9] introduces an iterative approach that generates a training dataset in the real-world domain using a pre-trained model and trains the model using the generated dataset. These methods usually rely on models trained on datasets designed manually, *e.g*., FlyingChairs and FlyingThings3D. Our work shows that using the self-supervised AutoFlow dataset can further improve performance and, more importantly, remove manual design processes from the entire pipeline.

Training datasets for optical flow. Due to the difficulty of constructing large-scale real-world annotated datasets for optical flow, synthetic data (*e.g*. FlyingChairs [6], FlyingThings3D [26], Kubric [8]) have been widely used as standard (pre-)training datasets. However, these datasets are generated without consideration of a target domain, so the domain gap always exists between the training and target domain, *e.g*., MPI Sintel [4] or VIPER [34] *vs*. KITTI [28].

Two works have introduced a training dataset generation pipeline based on real-world images. Depthstillation [2] synthesizes an image at an arbitrarily rotated view from a still image and provides optical flow ground truth between the images. RealFlow [9] synthesizes an intermediate frame between two frames given an estimated flow. The synthesis is controlled to have motion statistics similar to the target dataset. Both methods require off-the-shelf monocular depth methods [30, 31] and a hole-filling method to minimize artifacts on synthesized images. Furthermore, there is no guarantee that models trained on the synthesized datasets will perform optimally on the target domain.

AutoFlow [41] proposes a learning-to-render pipeline that learns dataset-rendering hyperparameters to optimize the optical flow accuracy on the target domain. Our method follows a similar direction, but unlike AutoFlow [41], does not require ground truth labels on the target domain. Instead, it uses a self-supervised search metric to update the rendering hyperparameters, making it applicable to any target domain without available ground truth.

3. Approach

Given an unlabeled target dataset $\mathbf{D}_{\text{target}}$, we aim to learn a synthetic dataset $\mathbf{D}_{\text{auto}}$ that approximately optimizes the performance in the target domain. To this end, we introduce self-supervised AutoFlow, which connects two independent research directions: (i) learning to render training datasets and (ii) self-supervised learning of optical flow (Sec. 3.2). Then, given the generated dataset $\mathbf{D}_{\text{auto}}$ with ground truth and the unlabeled target dataset $\mathbf{D}_{\text{target}}$, our method trains an optical flow network ϕ_θ using self-supervision to further adapt to the target domain (Sec. 3.3). The whole pipeline is fully self-supervised and does not require any ground truth optical flow from the target domain.

3.1. Preliminary: (Supervised) AutoFlow

AutoFlow [41] uses a layered approach to render a training dataset. The rendering pipeline uses a set of hyperparameters λ that control visual properties of foreground objects and the background (*e.g*. the number of moving objects, object shape, size, motion, *etc*.) and their visual effects (*e.g*. motion blur, fog, *etc*.) that appear in the rendered dataset. In a pre-defined hyperparameter search space Λ, an optimization process searches for an optimal set of hyperparameters λ^* such that $\phi_\theta(\lambda)$, an optical flow network trained on a rendered dataset with the parameters λ, minimizes a pre-defined search metric Ω on the target dataset:

$$\lambda^* = \operatorname*{argmin}_{\lambda \in \Lambda} \Omega\left(\phi_\theta(\lambda)\right). \tag{1}$$

AutoFlow [41] uses average end-point error (AEPE) for the search metric Ω that measures the accuracy between available ground truth in the target dataset and estimated optical flow from the trained model $\phi_\theta(\lambda)$. Despite promising results on Sintel and KITTI, AutoFlow cannot be applied to real-world data that do not have optical flow annotations.

3.2. Self-supervised AutoFlow

Motivation. To remove AutoFlow's dependence on in-domain ground truth, we look for inspiration from another line of research: self-supervised learning for optical flow. In particular, the recent SMURF [38] outperforms the supervised PWC-Net [42] (the state of the art 4 years ago) on Sintel and KITTI, suggesting that its self-supervised loss is highly correlated with the ground truth errors and could be a good proxy metric for learning optical flow.

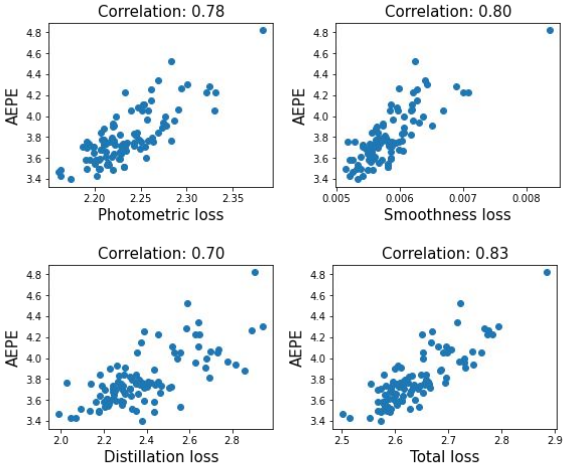


Figure 2. **Strong correlation** between ground truth error metric (AEPE) and self-supervised losses. We evaluate a set of RAFT models trained on supervised AutoFlow [41] datasets using the ground truth average end-point error (AEPE) and self-supervised losses averaged on the Sintel Final data. Each point in the plots corresponds to the performance of one model.

To this end, we analyze the correlation between the ground truth average end-point error (AEPE) metric and SMURF's [38] self-supervised loss on Sintel using the trained models during the hyperparameter search of the supervised AutoFlow [41], shown in Fig. 2. Each point in the plot corresponds to a RAFT model trained on a supervised AutoFlow dataset, with its AEPE on the Sintel Final split (y axis) and the self-supervised loss (x axis) that consists of a photometric loss, smoothness loss, and distillation loss. As shown in the plots, lower self-supervised losses correspond to lower AEPEs, and the correlation between the two signals increases when multiple losses are combined (*i.e*. total loss). This observation suggests that the self-supervised loss can also serve as a reliable proxy search metric and motivates our Self-supervised AutoFlow.

Self-supervised search metric. Our work extends the applicability of AutoFlow and presents Self-supervised AutoFlow (Self-AutoFlow or S-AF) which enables rendering a training dataset for a target domain by relying on the self-supervision loss metrics. We define our search metric Ω using a self-supervised loss which consists of three terms, a photometric loss $\mathcal{L}_{\text{photo}}$, a smoothness loss $\mathcal{L}_{\text{smooth}}$, and a distillation loss $\mathcal{L}_{\text{distill}}$,

$$\Omega_{\text{S-AF}}(\phi_\theta(\lambda)) = \mathcal{L}_{\text{photo}} + \omega_{\text{smooth}}\mathcal{L}_{\text{smooth}} + \omega_{\text{distill}}\mathcal{L}_{\text{distill}}, \tag{2}$$

where each loss function follows that of SMURF's [38] and ω_* are weighting coefficients. The input to each loss term is a pair of input images and an estimated optical flow from a trained model $\phi_\theta(\lambda)$, and are omitted for brevity.

The photometric loss $\mathcal{L}_{\text{photo}}$ penalizes the difference of corresponding pixels between input images $\mathbf{I}_t$ and $\mathbf{I}_{t+1}$.

$\mathbf{I}_{t+1}$ is differentiably warped into $\mathbf{I}_t$ using the predicted optical flow, $\mathbf{W}_t$, and following [51], a Hamming distance of ternary-census-transformed image patches of corresponding pixels is used to compute the photometric loss with respect to $\mathbf{W}_t$. The smoothness loss $\mathcal{L}_{\text{smooth}}$ uses the k^{th} order edge-aware smoothness to encourage continuity of the predicted optical flow field while allowing for discontinuity on edges. The distillation loss $\mathcal{L}_{\text{distill}}$ (*i.e.*, 'self-supervision loss' in SMURF [38]) applies a loss between a prediction on original images from a teacher model and a prediction on augmented and cropped images from a student model. As there is no backpropagation to the model in the search of AutoFlow, the search metric uses only the final, instead of all intermediate, flow prediction of RAFT.

Mixed datasets. Despite the high correlation between self-supervised loss and the ground truth error metric, there is no guarantee that the top candidate returned by self-supervised AutoFlow is the optimal set of hyperparameters according to the ground truth. To increase robustness, we choose the top-3 hyperparameter sets returned by self-supervised AutoFlow, generate a set of images with ground truth from each hyperparameter set, and mix them equally to form our final self-supervised AutoFlow dataset $\mathbf{D}_{\text{auto}}$. Empirically, we find that mixing the datasets decreases the likelihood of sampling a poor-performing AutoFlow hyperparameter and generally improves the robustness of the algorithm.

Discussion. There is a significant difference between learning a training set using self-supervised search metrics and self-supervised learning for optical flow. Self-supervised learning of optical flow involves training directly on a target dataset using self-supervised proxy losses. Gradients from the losses are directly backpropagated to update the model parameters. In contrast, our self-supervised AutoFlow approach optimizes hyperparameters for rendering a training dataset and trains the model on the dataset generated by the hyperparameters. The high correlation between the self-supervised loss and the ground truth error makes the Self-AutoFlow dataset almost as good as the AutoFlow dataset. The rendering pipeline can serve as an inductive bias for self-supervised learning and provide ground truth for complex scenes, such as occlusions and motion blur, that models trained on self-supervised losses tend to fail.

3.3. Combining Self-supervised AutoFlow with Self-supervised Optical Flow

Given the AutoFlow dataset $\mathbf{D}_{\text{auto}}$ learned from the self-supervised search metric, we further combine two data sources for training: (i) the self-supervised AutoFlow data $\mathbf{D}_{\text{auto}}$ and (ii) a target dataset without ground truth $\mathbf{D}_{\text{target}}$. Specifically, we first pre-train the model on $\mathbf{D}_{\text{auto}}$ and then self-supervised fine-tune the model on the target dataset $\mathbf{D}_{\text{target}}$, based on a training protocol from SMURF [38].

Self-supervised fine-tuning. This stage is to further adapt the model to the unlabeled target domain (*i.e.* raw videos). We use the same self-supervised loss from Eq. (2).

$$\mathcal{L} = \mathcal{L}_{\text{photo}} + \omega_{\text{smooth}}\mathcal{L}_{\text{smooth}} + \omega_{\text{distill}}\mathcal{L}_{\text{distill}}. \qquad (3)$$

Multi-frame fine-tuning. After fine-tuning on the target domain with the self-supervised loss in Eq. (3), we further apply the multi-frame fine-tuning from SMURF [38]. Given a triplet of input frames ($\mathbf{I}_{t-1}$, $\mathbf{I}_t$, and $\mathbf{I}_{t+1}$), SMURF predicts bi-directional flow $((\mathbf{I}_t \rightarrow \mathbf{I}_{t-1})$ and $(\mathbf{I}_t \rightarrow \mathbf{I}_{t+1}))$ and generates pseudo ground truth for the forward flow $\mathbf{W}_{\text{pseudo}}$ that includes more reliable estimation on occluded pixels through occlusion detection and inpainting using a shallow CNN. Then, we apply the following sequence loss from RAFT [43], which applies the l_1 loss (ρ_F) on each n^{th} intermediate output $\mathbf{W}^n$ with a decay factor γ,

$$\mathcal{L} = \sum_n \gamma^{N-n} \rho_F(\mathbf{W}_{\text{pseudo}} - \mathbf{W}^n). \qquad (4)$$

4. Experiments

4.1. Experimental setup

We use RAFT [43] as the backbone architecture. For the self-supervised hyperparameter search, we train 16 models in parallel using 96 NVIDIA P100 GPUs. At each search iteration, we train models for a short amount of steps, evaluate them on our search metric (Eq. (2)), and update the hyperparameters. We conduct 8 search iterations, which results in 16×8 total models for the search. We use the Adam optimizer (β_1=0.9, β_2=0.999) with a learning rate of 0.0001 and a one-cycle learning rate schedule [37]. Our method has the same theoretical complexity as AutoFlow. However, in practice, we reduce the computation cost by nearly 60% by using fewer training steps (80k).

For each target domain, we conduct a separate search and render a separate dataset. After pre-training on the rendered dataset, we further fine-tune the model with the self-supervised loss (Eq. (3)) on each target dataset, followed by multi-frame fine-tuning (Eq. (4)). We use a learning rate of 0.0002 with an exponential decay during the last 20% of steps. At inference time, we use the established evaluation scheme for each domain. Tab. 1 follows AutoFlow's, which uses a fixed resolution during inference (Sintel: 448×1024, KITTI: 640×640). Tab. 3 and Tab. 4 follow SMURF's, which uses resolutions that perform the best on the training set (Sintel: 384×1024, KITTI: 424×952).

4.2. Self-supervised AutoFlow

Comparison with the state-of-the-art pre-training approaches. Tab. 1 compares our method with different pre-training approaches and reports the accuracy on Sintel and

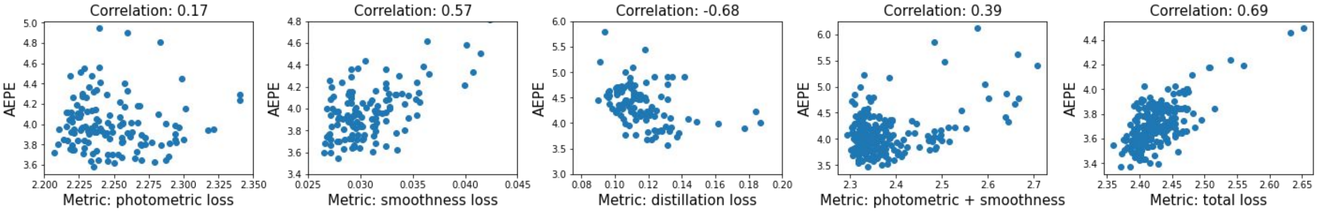


Figure 3. **Ablation study of self-supervised search metric.** None of the individual terms of the standard self-supervised loss, when used as a search metric, is strongly correlated with AEPE on the target dataset. Only the combination of all three terms leads to a strong correlation between the search metric and the AEPE. Each point here denotes an AEPE of a model trained on a generated dataset searched by a self-supervised search metric, whereas Fig. 2 shows the supervised AutoFlow models that use AEPE for the dataset parameter search.

Table 1. **Comparison of (self-)supervised pre-training approaches.** Our Self-AutoFlow (S-AF) outperforms FlyingChairs pre-training and is competitive with supervised AutoFlow (AF) which is learned from ground truth annotations. **Bold** indicates the best number. "AF X", "AF-mix X" or "S-AF X" indicates that AF or S-AF is learned for the dataset X. Numbers in parentheses indicate the number of training steps.

Dataset and Method	Sintel Clean (AEPE ↓)	Sintel Final (AEPE ↓)	KITTI (AEPE ↓)
Supervised			
RAFT Chairs [43]	2.27	3.76	7.63
AF Sintel (3.2M) [40]	**1.74**	**2.41**	4.18
AF-mix Sintel (3.2M)	1.85	2.53	3.92
AF KITTI (0.8M) [41]	2.09	2.82	4.33
AF-mix KITTI (0.8M)	1.87	2.77	**3.86**
Self-supervised			
SMURF Chairs [38]	2.19	3.35	7.94
S-AF Sintel (3.2M)	**1.83**	**2.59**	5.22
S-AF KITTI (0.2M)	2.20	3.01	4.58
S-AF KITTI (0.8M)	1.99	3.00	4.29
S-AF KITTI (3.2M)	1.88	2.85	**4.22**

Table 2. **Ablation study on end-to-end training.** The models are trained with the dataset-mixing strategy and longer training steps.

	$\mathcal{L}_{\text{photo}}$	$\mathcal{L}_{\text{smooth}}$	$\mathcal{L}_{\text{distill}}$	$\mathcal{L}_{\text{photo}} + \mathcal{L}_{\text{smooth}}$	$\mathcal{L}_{\textbf{total}}$
Sintel Clean	2.26	2.18	2.07	2.30	**1.83**
Sintel Final	3.24	2.84	3.04	2.98	**2.59**

KITTI. All methods use RAFT [43] as the backbone architecture. AutoFlow (AF) [41] and our Self-AutoFlow (S-AF) are trained on each rendered dataset for Sintel or KITTI, and we report accuracy on both benchmark datasets. S-AF mixes rendered datasets from top-3 hyperparameter sets that show low metric score (see Sec. 3.2); for a fair comparison, we prepare an equivalent model for AutoFlow and denote it as AF-mix. "AF X", "AF-mix X" or "S-AF X" indicates that AutoFlow (AF) or self-supervised AutoFlow (S-AF) is learned for the target domain X. Our dataset-mixing strategy improves AF-KITTI from their reported number 4.33 to 3.86, demonstrating its effectiveness for both supervised and self-supervised setups.

Our method substantially outperforms (self-)supervised pre-trained models on FlyingChairs and performs competitively to (supervised) AutoFlow and AutoFlow-mix. The performance gap between S-AF KITTI and AF KITTI (4.29 *vs*. 3.86) is much smaller than that between Chairs and AF (7.63 *vs*. 3.86). We note that the accuracy in Tab. 1 is reported on the training set, where AF uses its ground truth to optimize, and thus is guaranteed to outperform S-AF. It is significant to achieve such a small performance gap, suggesting that our self-supervised approach can successfully extend the applicability of AutoFlow on unlabeled target domains as demonstrated in Sec. 4.5.

Ablation study of self-supervised search metric. Fig. 3 provides an ablation study on our search metrics in Eq. (2). Similar to Fig. 2, each data point corresponds to a trained model with its AEPE on Sintel Final (y axis) and a loss value on a metric (x axis) that is used for our S-AF hyperparameter search to render its training dataset.

Unlike in Fig. 2 where we observe a strong correlation between the supervised search metric (AEPE) and the measured self-supervised loss, here we observe very different behavior. Each of the individual self-supervised signals performs poorly as a search metric, when judged by the AEPE of the models trained on rendered datasets that are searched by the self-supervised signals. For example, a S-AF hyperparameter search guided by the distillation loss converges to models with very high AEPE but low distillation loss because distillation alone can lead to trivial solutions, such as a model predicting zero or constant flow for any input. As a result, only the combination of all three self-supervised signals act as an effective search metric, showing the highest correlation with AEPE and the lowest AEPE (< 3.4).

Tab. 2 reports the AEPE of models trained on rendered datasets optimized for different self-supervised metrics. Note, the models in this table use the full training setup, including the dataset-mixing strategy and longer training steps. The model with $\mathcal{L}_{\text{total}}$ shows the lowest AEPE, confirming that the combination of three losses serves as a reliable search metric.

4.3. Self-supervised Learning of Optical Flow

Comparison to the state of the art. In Sec. 3.3, we combine our S-AF (Sec. 3.2) with the self-supervised learning

Table 3. **Comparison of self-supervised learning approaches.** Our models are pre-trained on self-supervised AutoFlow (S-AF) and the self-supervised objective (SS) using unlabeled data from the target dataset. Following SMURF [38], we train two models for each dataset on either the training split or the test split and evaluate on the other, denoted as **S-AF+SS train** and **S-AF+SS test**. Our method performs favorably against the state of the art. "{}" trained on/using the unlabeled evaluation set; "[]" trained on data closed to evaluation set; "MF" using multi-frame estimation at test time [38].

	Sintel Clean [4]		Sintel Final [4]		KITTI 2015 [28]			
	AEPE ↓		AEPE ↓		AEPE ↓	AEPE (noc) ↓	Fl-all (%) ↓	
Method	*train*	*test*	*train*	*test*	*train*	*train*	*train*	*test*
EPIFlow [53]	3.94	7.00	5.08	8.51	5.56	2.56	–	16.95
UFlow [20]	3.01	5.21	4.09	6.50	2.84	1.96	9.39	11.13
SemiFlow [16]	**1.30**	–	2.46	–	3.35	–	11.12	–
SMURF test [38]	1.99	–	2.80	–	2.01	1.42	6.72	–
S-AF+SS test	1.65	–	**2.40**	–	**1.94**	**1.37**	**6.56**	–
DDFlow [24]	{2.92}	6.18	{3.98}	7.40	[5.72]	[2.73]	–	14.29
SelFlow [25] (MF)	[2.88]	[6.56]	{3.87}	{6.57}	[4.84]	[2.40]	–	14.19
UnsupSimFlow [15]	{2.86}	5.92	{3.57}	6.92	[5.19]	–	–	[13.38]
ARFlow [23] (MF)	{2.73}	{4.49}	{3.69}	{5.67}	[2.85]	–	–	[11.79]
RealFlow [9]	{1.34}	–	{2.38}	–	{2.16}	–	–	–
SMURF train [38]	{1.71}	3.15	{2.58}	4.18	{2.00}	{1.41}	{6.42}	6.83
S-AF+SS train	{1.51}	**3.03**	{2.30}	**3.98**	{1.96}	{1.38}	{6.26}	**6.76**

Table 4. **Generalization across datasets.** We compare the generalization ability of self-supervised optical flow methods. We train the models on one dataset and evaluate on others. Our method (S-AF) outperforms SMURF on cross-dataset evaluations. SS Sintel/KITTI means further self-supervised training on Sintel/KITTI.

	Chairs	Sintel *train*		KITTI-15 *train*	
Method	*test*	Clean	Final	AEPE	Fl-all (%)
SMURF Chairs	1.72	2.19	3.35	7.94	26.51
S-AF Sintel	**1.61**	**1.83**	**2.57**	4.79	15.47
S-AF KITTI	2.09	2.16	2.96	**4.28**	**13.60**
+ SS Sintel					
SMURF	1.99	1.99	2.80	4.47	12.55
S-AF	**1.81**	**1.65**	**2.40**	**4.28**	**12.45**
+ SS KITTI					
SMURF	3.26	3.38	4.47	2.01	6.72
S-AF	**3.19**	**3.32**	**4.44**	**1.94**	**6.56**

Table 5. **Supervised fine-tuning on public benchmarks.** We fine-tune our model using ground truth in a supervised manner. (AEPE ↓ for Sintel and Fl-all ↓ for KITTI. Methods using warm start on Sintel are marked by *). Models pre-trained on self-supervised AutoFlow (S-AF) can serve as a good initialization for supervised fine-tuning.

Method	Sintel Clean	Sintel Final	KITTI
RealFlow [9]	-	-	4.63 %
SemiFlow (RAFT)* [16]	1.65	2.79	4.85 %
RAFT-it [40]	1.55	2.90	4.31 %
RAFT-S-AF	**1.42**	**2.75**	**4.12 %**

approach of optical flow to further adapt the model to the target domains, denoted by **S-AF+SS**. We compare against state-of-the-art approaches that do not use ground truth in the target domain in Tab. 3. We train our model on the standard train/test splits for Sintel and further train on the multi-view extension data following [38] for the KITTI dataset. We train two models for each dataset, one trained on the test split (* test) in a self-supervised manner and evaluated on the training split with ground truth, and the other trained on the training split (* train) and evaluated on the test split (*i.e.*, benchmark websites).

Compared to SMURF, our method reduces the AEPE by 0.12 on Sintel Clean test, 0.20 on Sintel Final test, and F1-all by 0.07 on KITTI test. Our method is comparable to SemiFlow [16] and RealFlow [9] on Sintel Clean train, although both SemiFlow and RealFlow are pre-trained on FlyingChairs and FlyingThings3D and thus have strong performance on Sintel Clean, due to the proximity of their domains. Our method outperforms SemiFlow and RealFlow on the more challenging Sintel Final train and KITTI.

Generalization across datasets. In Tab. 4, we evaluate the generalization of our approach by training the model on one dataset and evaluating it on other datasets. We denote the models with self-supervised fine-tuning on target datasets as +SS Sintel/KITTI. When only training on S-AF datasets, the model achieves an AEPE of 1.83 on Sintel Clean and 2.57 on Sintel Final, which outperforms SMURF with self-supervised fine-tuning on the target Sintel dataset by 0.16 and 0.23. Both models trained on S-AF and self-supervised fine-tuned on Sintel/KITTI achieve the best cross-domain performance on all the target datasets.

4.4. Supervised Fine-tuning on Public Benchmarks

To examine how well our method can serve as a good initialization, we fine-tune our S-AF+SS train model in Tab. 3 using the same fine-tuning protocol from [40]. As shown in Tab. 5, our method consistently outperforms RAFT-it [40], SemiFlow, and RealFlow, indicating that S-AF+SS models can serve as a good initialization for supervised fine-tuning.

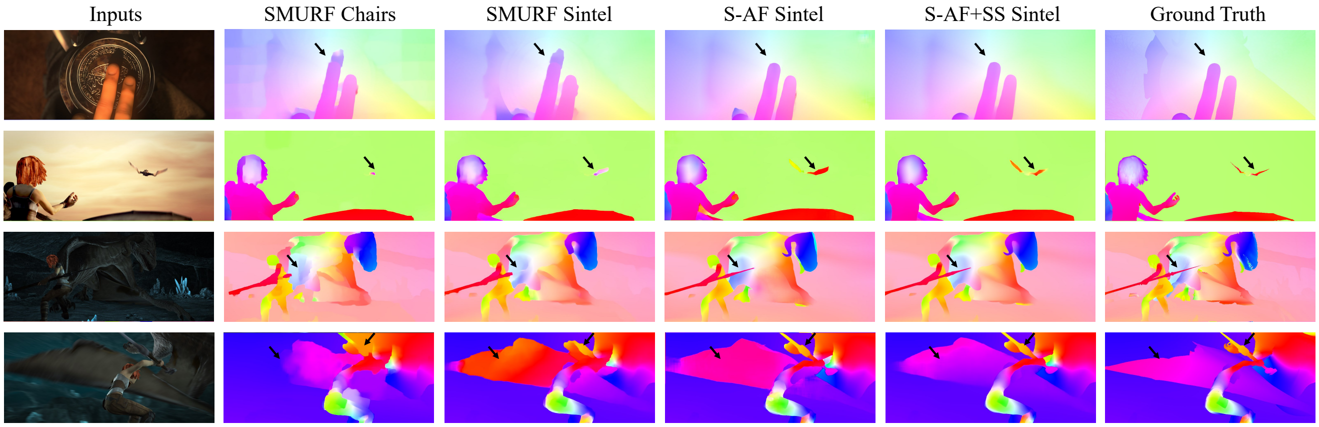


Figure 4. **Comparison of self-supervised methods on Sintel.** SMURF, both pre-trained (SMURF Chairs) and self-supervised fine-tuned (SMURF Sintel), tends to fail on shadows, strong motion blur, or small/thin objects. On the other hand, self-supervised AutoFlow (S-AF) on Sintel provides more reliable predictions, and self-supervised (SS) fine-tuning (S-AF+SS Sintel) further improves the results.

4.5. Evaluation on Downstream Tasks

To further demonstrate the generalization of our method to a real-world domain without ground truth, we compare our method with various supervised, semi-supervised, and self-supervised methods on two downstream tasks: keypoint propagation and segmentation tracking on the DAVIS dataset [29]. For self-supervised fine-tuning on DAVIS, we use the seven BADJA sequences and three challenging sequences (drift-turn, drift-chicane and color-run) as the test set, and the remaining 80 sequences for training.

Keypoint propagation. For evaluation, we use the Percentage of Correct Keypoint-Transfer (PCK-T) metric [47] with keypoint annotations from the BADJA dataset [3]. Given annotated keypoints on a reference image, the metric calculates the percentage of correctly propagated keypoints along a video sequence. As shown in Tab. 6, our S-AF DAVIS model achieves better accuracy than other (semi-)supervised approaches (SemiFlow, RealFlow, AF Sintel, and RAFT-it) and a self-supervised pre-training approach (SMURF Chairs). Our self-supervised fine-tuned model on DAVIS (S-AF+SS DAVIS) outperforms SMURF DAVIS.

Compared to the S-AF results that use different unlabeled data as target, S-AF DAVIS outperforms S-AF Sintel and KITTI, showing that our method successfully learns a better dataset for the target domains without using the ground truth labels.

Segmentation tracking. We propagate initial segmentation masks using optical flow and evaluate IoU between the propagated and ground truth masks. As shown in Tab. 7, Our method (S-AF) consistently outperforms supervised AutoFlow, RAFT, and SMURF. Since the performance difference is mainly on tiny objects or around object boundaries, the $>1\%$ difference between S-AF+SS Davis and SMURF Davis is a moderate improvement.

Table 6. **Keypoint propagation on the BADJA dataset [3].** We use different optical flow methods to propagate the keypoints along the sequences and report the PCK-T metric. (S-)AF: (self-supervised) AutoFlow; SS: self-supervised fine-tuning. SMURF DAVIS is first trained on Chairs and then fine-tuned on DAVIS.

Method	bear	camel	cows	dog-a	dog	horse-h	horse-l	Avg.
DINO [5]	75.7	58.2	71.4	10.3	46.0	35.8	56.5	50.6
PIPs [10]	76.3	84.0	79.1	31.6	42.9	60.4	58.6	**61.8**
(Semi-)supervised								
SemiFlow-Davis [16]	66.4	72.0	71.4	13.8	40.8	36.4	31.4	47.5
RealFlow-Davis [9]	64.3	80.1	63.4	10.3	45.4	32.5	38.7	47.8
AF Sintel [41]	71.4	80.1	75.1	17.2	47.1	34.4	27.2	50.4
RAFT-it [40]	73.2	83.0	78.1	17.2	46.0	39.1	30.4	**52.4**
Pre-training								
SMURF Chairs [38]	79.3	74.0	73.8	3.4	42.5	34.4	29.3	48.1
S-AF Sintel	73.2	83.9	62.0	3.4	42.0	40.4	26.7	47.4
S-AF KITTI	72.5	76.8	73.8	0.0	46.6	34.4	31.9	48.0
S-AF DAVIS	72.9	76.5	75.7	20.7	47.7	38.4	31.4	**51.9**
Self-supervised fine-tuning								
SMURF DAVIS [38]	80.0	83.0	77.8	3.4	47.1	40.4	44.0	53.7
S-AF+SS DAVIS	80.0	82.3	74.9	10.3	50.6	43.0	42.4	**54.8**

Table 7. **Segmentation tracking on DAVIS.** We propagate the initial segmentation masks using optical flow and evaluate IoU compared to ground truth masks.

AF Sintel	RAFT-it	SMURF Chairs	**S-AF Davis**	SMURF Davis	**S-AF+SS Davis**
0.830	0.801	0.807	**0.837**	0.876	**0.888**

4.6. Visual Comparison

Sintel. As in Fig. 4, compared to out-of-domain pre-training approaches, S-AF Sintel performs better than SMURF Chairs on shadows, small/thin objects and scenes with strong motion blur. Self-supervised fine-tuning on Sintel (S-AF+SS Sintel model) further improves the results upon the pre-training S-AF Sintel model, whereas SMURF still tends to fail on those cases. The results show that our self-supervised learning-to-render approach not only pro-

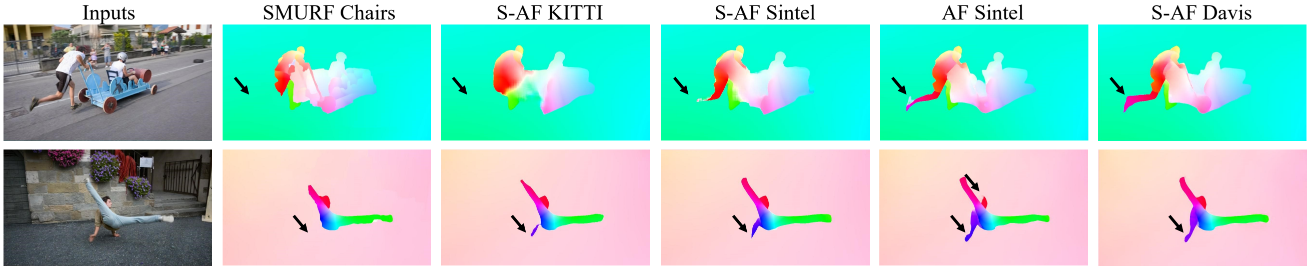


Figure 5. **Visual comparison of pre-training on DAVIS.** Compared to SMURF Chairs and AutoFlow (AF) Sintel, self-supervised AutoFlow (S-AF) DAVIS learned from DAVIS data yields better flow results. In addition, S-AF DAVIS outperforms S-AF Sintel and S-AF KITTI, indicating S-AF successfully learns a better training set to adapt the model to a target domain.

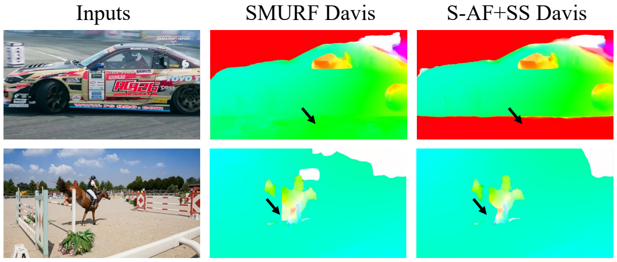


Figure 6. **Visual comparison of self-supervised fine-tuning on DAVIS.** With self-supervised fine-tuning on the target DAVIS dataset, our S-AF+SS DAVIS predicts more accurate flow for textureless areas or thin objects, showing that the better initialization of S-AF leads to better self-supervised fine-tuning results compared to the initialization from pre-training on FlyingChairs.

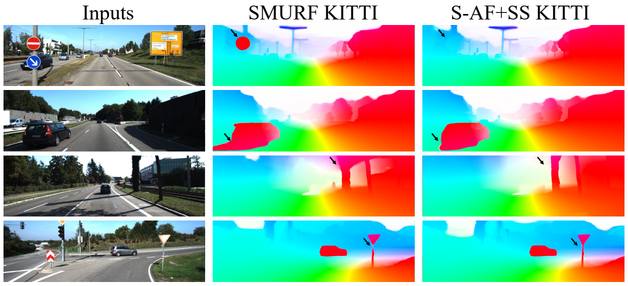


Figure 7. **Visual comparison of self-supervised fine-tuning on KITTI.** Our S-AF+SS KITTI model predicts more accurate flow on objects with large motion, on shadows, and on thin structures of the scene compared to SMURF KITTI. Purely self-supervised methods may predict incorrect flow fields, while our S-AF+SS approach resolves this issue by a better initialization.

vides a strong pre-trained model on the target domain, but also serves as a good initialization for the self-supervised fine-tuning; suggesting that our S-AF is complementary to the self-supervised learning approach.

DAVIS. Fig. 5 shows the comparison of different pre-training methods on the DAVIS dataset. The SMURF Chairs model does not clearly capture the motion of the foot and hand of the person due to the domain gap between FlyingChairs and DAVIS. AF Sintel does not generalize well to the real-world DAVIS data; our S-AF learned from the unlabeled DAVIS data successfully captures the detailed structure. We further compare our models that use different target domains for dataset generation (S-AF DAVIS, S-AF Sintel, S-AF KITTI). S-AF DAVIS shows the best results by successfully optimizing the rendering parameters for the real-world target domain, *i.e*. DAVIS.

Fig. 6 shows that self-supervised fine-tuning on DAVIS (S-AF+SS DAVIS) further improves the result over SMURF DAVIS, showing that the better initialization of S-AF leads to better self-supervised fine-tuning results compared to the initialization from pre-training on FlyingChairs.

KITTI. As shown in Fig. 7, S-AF+SS KITTI predicts more accurate flow on close objects with large motion, shadows, and thin structures of the scene than SMURF KITTI. The results suggest that the purely self-supervised method may predict incorrect flow due to optimizing the photometric constancy loss. In contrast, our S-AF pre-training approach provides a better model initialization and resolves this issue by pre-training on rendered S-AF data ground truth.

Discussions. Despite the promising results, the visual comparison suggests that there is room for improvement, such as the thin structures in Fig. 5 and the sky regions in Fig. 6. Future work may further explore using a more realistic rendering engine *e.g*., with a sky model, and developing better self-supervised losses to address these issues.

5. Conclusions

We have introduced self-supervised AutoFlow to learn a training set for optical flow for unlabeled data using self-supervised metrics. Self-supervised AutoFlow performs on par with AutoFlow that uses ground truth on Sintel and KITTI, and better on the real-world DAVIS dataset where ground truth is not available. Our work suggests the benefits of connecting learning to render with self-supervision and we hope to see more work in this direction to solve optical flow in the real world.

References

[1] Aria Ahmadi and Ioannis Patras. Unsupervised convolutional neural networks for motion estimation. In *ICIP*, 2016. 2

[2] Filippo Aleotti, Matteo Poggi, and Stefano Mattoccia. Learning optical flow from still images. In *CVPR*, 2021. 2

[3] Benjamin Biggs, Thomas Roddick, Andrew Fitzgibbon, and Roberto Cipolla. Creatures great and SMAL: Recovering the shape and motion of animals from video. In *ACCV*, 2018. 7

[4] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. 1, 2, 6

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 7

[6] 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. In *ICCV*, 2015. 1, 2

[7] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 2

[8] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J. Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *CVPR*, 2022. 2

[9] Yunhui Han, Kunming Luo, Ao Luo, Jiangyu Liu, Haoqiang Fan, Guiming Luo, and Shuaicheng Liu. RealFlow: EM-based realistic optical flow dataset generation from videos. In *ECCV*, 2022. 2, 6, 7

[10] Adam W. Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *ECCV*, 2022. 7

[11] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. FlowFormer: A transformer architecture for optical flow. In *ECCV*, 2022. 2

[12] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. LiteFlowNet: A lightweight convolutional neural network for optical flow estimation. In *CVPR*, 2018. 2

[13] Clive Humby. Data is the new oil. *Proc. ANA Sr. Marketer's Summit. Evanston, IL, USA*, 2006. 1

[14] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017. 2

[15] Woobin Im, Tae-Kyun Kim, and Sung-Eui Yoon. Unsupervised learning of optical flow with deep feature similarity. In *ECCV*, 2020. 1, 2, 6

[16] Woobin Im, Sebin Lee, and Sung-Eui Yoon. Semi-supervised learning of optical flow by flow supervisor. In *ECCV*, 2022. 2, 6, 7

[17] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver IO: A general architecture for structured inputs & outputs. In *ICLR*, 2022. 2

[18] Azin Jahedi, Lukas Mehl, Marc Rivinius, and Andrés Bruhn. Multi-Scale RAFT: Combining hierarchical concepts for learning-based optical flow estimation. In *ICIP*, 2022. 2

[19] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *ICCV*, 2021. 2

[20] Rico Jonschkowski, Austin Stone, Jonathan T. Barron, Ariel Gordon, Kurt Konolige, and Anelia Angelova. What matters in unsupervised optical flow. In *ECCV*, 2020. 2, 6

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 1

[22] Wei-Sheng Lai, Jia-Bin Huang, and Ming-Hsuan Yang. Semi-supervised learning for optical flow with generative adversarial networks. In *NIPS*, 2017. 2

[23] Liang Liu, Jiangning Zhang, Ruifei He, Yong Liu, Yabiao Wang, Ying Tai, Donghao Luo, Chengjie Wang, Jilin Li, and Feiyue Huang. Learning by Analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In *CVPR*, 2020. 1, 2, 6

[24] Pengpeng Liu, Irwin King, Michael R. Lyu, and Jia Xu. DDFlow: Learning optical flow with unlabeled data distillation. In *AAAI*, 2019. 1, 2, 6

[25] Pengpeng Liu, Michael R. Lyu, Irwin King, and Jia Xu. SelFlow: Self-supervised learning of optical flow. In *CVPR*, 2019. 1, 2, 6

[26] 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. 1, 2

[27] Simon Meister, Junhwa Hur, and Stefan Roth. UnFlow: Unsupervised learning of optical flow with a bidirectional census loss. In *AAAI*, 2018. 2

[28] Moritz Menze, Christian Heipke, and Andreas Geiger. Joint 3D estimation of vehicles and scene flow. In *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015. 1, 2, 6

[29] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 2, 7

[30] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021. 2

[31] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *PAMI*, 2020. 2

[32] Anurag Ranjan and Michael J. Black. Optical flow estimation using a spatial pyramid network. In *CVPR*, 2017. 2

[33] Zhe Ren, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Yang, and Hongyuan Zha. Unsupervised deep learning for optical flow estimation. In *AAAI*, 2017. 2

[34] Stephan R. Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *ICCV*, 2017. 2

[35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 2

[36] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 1

[37] Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019. 4

[38] Austin Stone, Daniel Maurer, Alper Ayvaci, Anelia Angelova, and Rico Jonschkowski. SMURF: Self-teaching multi-frame unsupervised raft with full-image warping. In *CVPR*, 2021. 2, 3, 4, 5, 6, 7

[39] Xiuchao Sui, Shaohua Li, Xue Geng, Yan Wu, Xinxing Xu, Yong Liu, Rick Goh, and Hongyuan Zhu. CRAFT: Cross-attentional flow transformer for robust optical flow. In *CVPR*, 2022. 2

[40] Deqing Sun, Charles Herrmann, Fitsum Reda, Michael Rubinstein, David J. Fleet, and William T. Freeman. Disentangling architecture and training for optical flow. In *ECCV*, 2022. 2, 5, 6, 7

[41] Deqing Sun, Daniel Vlasic, Charles Herrmann, Varun Jampani, Michael Krainin, Huiwen Chang, Ramin Zabih, William T. Freeman, and Ce Liu. AutoFlow: Learning a better training set for optical flow. In *CVPR*, 2021. 1, 2, 3, 5, 7

[42] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 2, 3

[43] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 2, 4, 5

[44] Yang Wang, Yi Yang, Zhenheng Yang, Liang Zhao, Peng Wang, and Wei Xu. Occlusion aware unsupervised learning of optical flow. In *CVPR*, 2018. 2

[45] Taihong Xiao, Jinwei Yuan, Deqing Sun, Qifei Wang, Xin-Yu Zhang, Kehan Xu, and Ming-Hsuan Yang. Learnable cost volume using the cayley representation. In *ECCV*, 2020. 2

[46] Haofei Xu, Jiaolong Yang, Jianfei Cai, Juyong Zhang, and Xin Tong. High-resolution optical flow from 1D attention and correlation. In *ICCV*, 2021. 2

[47] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *PAMI*, 35(12):2878–2890, 2012. 7

[48] Yanchao Yang and Stefano Soatto. Conditional prior networks for optical flow. In *ECCV*, 2018. 2

[49] Jason J. Yu, Adam W. Harley, and Konstantinos G. Derpanis. Back to Basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *ECCVW*, 2016. 2

[50] Shuai Yuan, Xian Sun, Hannah Kim, Shuzhi Yu, and Carlo Tomasi. Optical flow training under limited label budget via active learning. In *ECCV*, 2022. 2

[51] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *ECCV*, 1994. 4

[52] Feihu Zhang, Oliver J. Woodford, Victor Adrian Prisacariu, and Philip H.S. Torr. Separable flow: Learning motion cost volumes for optical flow estimation. In *ICCV*, 2021. 2

[53] Yiran Zhong, Pan Ji, Jianyuan Wang, Yuchao Dai, and Hongdong Li. Unsupervised deep epipolar flow for stationary or dynamic scenes. In *CVPR*, 2019. 1, 2, 6