Self-supervised Learning for Video Correspondence Flow

Zihang Lai
zihang.lai@cs.ox.ac.uk
Weidi Xie
weidi@robots.ox.ac.uk

Department of Computer Science
University of Oxford, UK
Visual Geometry Group,
Department of Engineering Science
University of Oxford, UK

Abstract

The objective of this paper is *self-supervised learning* of feature embeddings that are suitable for matching correspondences along the videos, which we term correspondence flow. By leveraging the natural spatial-temporal coherence in videos, we propose to train a "pointer" that reconstructs a target frame by copying pixels from a reference frame.

We make the following contributions: *First*, we introduce a simple information bottleneck that forces the model to learn robust features for correspondence matching, and prevent it from learning trivial solutions, *e.g.* matching based on low-level colour information. *Second*, to alleveate tracker drifting, due to complex object deformations, illumination changes and occlusions, we propose to train a recursive model over long temporal windows with scheduled sampling and cycle consistency. *Third*, we achieve the state-of-the-art performance on DAVIS 2017 video segmentation and JHMDB keypoint tracking tasks, outperforming all previous self-supervised learning approaches by a significant margin. *Fourth*, in order to shed light on the potential of self-supervised learning on the task of video correspondence flow, we probe the upper bound by training on additional data, *i.e.* more diverse videos, further demonstrating significant improvements on video segmentation. The source code will be released at https://github.com/zlai0/CorrFlow.

1 Introduction

Correspondence matching is a fundamental building block for numerous applications ranging from depth estimation [24] and optical flow [4, 8, 15], to segmentation and tracking [37], and 3D reconstruction [12]. However, training models for correspondence matching is not trivial, as obtaining manual annotations can be prohibitively expensive, and sometimes is even impossible due to occlusions and complex object deformations. In the recent works [35, 36], Rocco *et al*. proposed to circumvent this issue by pre-training Convolutional Neural Networks (CNNs) for predicting artificial transformations, and further bootstrap the model by finetuning on a small dataset with human annotations. Alternatively, the necessity for labbelled data can be avoided by using self-supervised learning, *i.e*. a form of unsupervised learning, where part of the data is withheld for defining a proxy task, such that the model will be forced to learn the semantic representation that we really care about.

(a) DAVIS 2017 Video Segmentation (b) Keypoint Tracking

Figure 1: We propose self-supervised learning of correspondence flow on videos. Without any fine-tuning, the acquired representation generalizes to various tasks: (a) video segmentaiton; (b) keypoint tracking. **Note**: For both tasks, the annotation for the first frame is given, the goal is to propagate the annotations through the videos.

Videos have shown to be appealing as a data source for self-supervised learning due to their almost infinite supply (from YouTube etc), and the availability of numerous proxy losses that can be employed from the intrinsic spatio-temporal coherence, *i.e*. the signals in video tend to vary smoothly in time [19, 49]. In this paper, we propose to tackle the task of correspondence flow in videos with self-supervised learning. We are certainly not the first to explore this idea, in the seminal paper by Vondrick *et al*. [40], they propose to learn embeddings for grayscale frames and use *colorization* as a proxy task for training. Despite the promising efforts, the model capacity has been significantly constrained due to the loss of colour information, and suffers the problem of tracker drifting as only pair of frames are used during training.

We make the following contributions: *First*, we introduce an embarrassingly simple idea to avoid trivial solutions while learning pixelwise matching by frame reconstruction. During training, channel-wise dropout and colour jitterings are added intentionally on the input frames, the model is therefore forced *not* to rely on low-level colour information, and must be robust to colour jittering. *Second*, we propose to train the model recursively on videos over long temporal windows with scheduled sampling and forward-backward consistency. Both ideas have shown to improve the model robustness and help to alleviate the tracker drifting problem. *Third*, after self-supervised training, we benchmark the model on two downstream tasks focusing on pixelwise tracking, *e.g*. DAVIS 2017 video segmentation and JHMDB keypoint tracking, outperforming all previous self-supervised learning approaches by a significant margin. *Fourth*, to further shed light on the potential of self-supervised learning for video correspondence flow, we probe the upper bound by training on more diverse video data, and further demonstrating significant improvements on video segmentation.

2 Related Work

Correspondence Matching. Recently, many researchers have studied the task of correspondence matching using deep Convolutional Neural Networks [11, 26, 31, 35, 36]. The works from Rocco *et al*. [35, 36], propose to train CNNs by learning the artificial transformations between pairs of images. For robust estimation, they applied a differentiable soft inlier score for evaluating the quality of alignment between spatial features and providing a loss for learning semantic correspondences. However, their work may not be ideal as the model

still relies on synthetic transformations. In contrast, we address the challenge of learning correspondence matching by exploiting the temporal coherence in videos.

Optical Flow. In the conventional variational approaches, optical flow estimation is treated as an energy minimization problem based on brightness constancy and spatial smoothness [14]. In later works, feature matching is used to firstly establish sparse matchings, and then interpolated into dense flow maps in a pyramidal coarse-to-fine manner [5, 34, 47]. Recently, convolutional neural networks (CNNs) have been applied to improve the matching by learning effective feature embeddings [2, 21]. Another line of more relevant research is unsupervised learning for optical flow. The basic principles are based on brightness constancy and spatial smoothness [29, 45, 52]. This leads to the popular photometric loss which measures the difference between the reference image and the warped image. For occluded regions, a mask is implicitly estimated by checking forward-backward consistency.

Self-supervised Learning. Of more direct relevance to our training framework are self-supervised frameworks that use video data [1, 7, 9, 10, 16, 17, 18, 19, 20, 22, 25, 27, 30, 40, 44, 48]. In [9, 30, 46], the proxy task is defined to focus on temporal sequence ordering of the frames. Another approach is to use the temporal coherence as a proxy loss [16, 19, 44]. Other approaches use egomotion [1, 18] in order to enforce equivariance in feature space [18]. Recently [40], leveraged the natural temporal coherency of colour in videos, to train a network for tracking and correspondence related tasks. Our approach builds in particular on those that use frame synthesis [7, 20, 48], though for us synthesis is a proxy task rather than the end goal.

3 Approach

The goal of this work is to train an embedding network with self-supervised learning that enables pixelwise correspondence matching. Our basic idea is to exploit spatial-temporal coherence in videos, that is, the frame appearances will not change abruptly, and colours can act as a reliable supervision signal for learning correspondences.

3.1 Background

In this section, we briefly review the work by Vondrick *et al.* [40]. Formally, let $c_i \in \mathbb{R}^d$ be the true colour for pixel i in the reference frame, and let $c_j \in \mathbb{R}^d$ be the true colour for a pixel j in the target frame. $y_j \in \mathbb{R}^d$ is the model's prediction for c_j, it is a linear combination of colours in the reference frame:

$$y_j = \sum_i A_{ij} c_i, \qquad \text{where } A_{ij} = \frac{exp\langle f_i^T f_j \rangle}{\sum_k exp\langle f_k^T f_j \rangle} \tag{1}$$

A is an affinity matrix computed from simple dot products between the feature embeddings of the *grayscale* target and reference frame (f's).

Despite the efforts in circumventing trivial solutions, training models with *grayscale* inputs has introduces a train-test discrepancy when deploying to the downstream task, as the model has never been trained to encode the correlation of RGB channels. Another issule lies in the fact that their model is only trained with pairs of ground truth video frames, which inevitably leads to model drifting when evaluated on video tracking tasks. As the prediction for later steps rely on the prediction from previous steps, the errors accumulate, and there is no mechanism for the model to recover from previous error states.

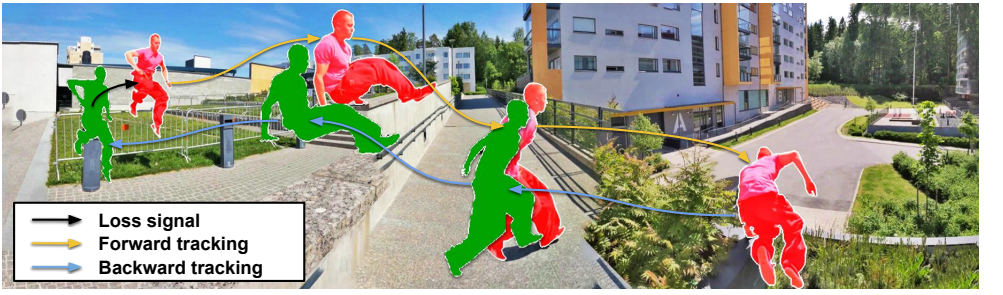


Figure 2: An overview of the proposed self-supervised learning for correspondence flow. A recursive model is used to compute the dense correspondence matching over a long temporal window with forward-backward cycle consistency.

In the following sections, we propose to train a framework that aims to close the gap between training and testing as much as possible, *i.e.* the model should ideally be trained on *full-colour* and *high-resolution* over *long* video sequences.

3.2 Feature Embedding with Information Bottleneck

Given a collection of frames $\{I_1, I_2, ..., I_N\}$ from a video clip, we parametrize the feature embedding module with CNNs:

$$f_i = \Phi(g(I_i); \theta) \tag{2}$$

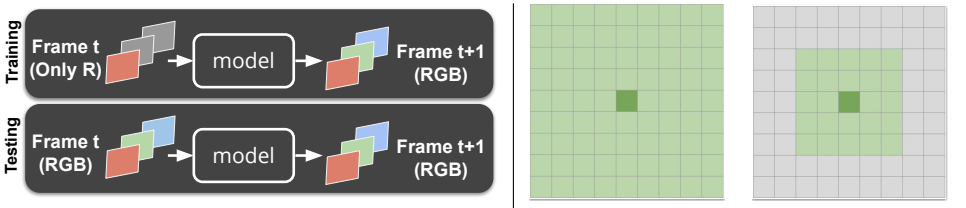
where Φ refers to a ResNet feature encoder (details in Arxiv paper[1]), and $g(\cdot)$ denotes an *information bottleneck* that prevents from information leakage. In Vondrick *et al*. [40], this is simply defined as a function that converts RGB frame into *grayscale*, therefore, the model is forced *not* to rely on colours for matching correspondences.

We propose to randomly zero out 0, 1, or 2 channels in each input frame with some probability (one possible input is shown in Figure 3 (a)), and perturb the brightness, contrast and saturation of an image by up to 10%. Despite the simplicity, this design poses two benefits: *First*, the input jitterings and stochastic dropout are essentially acting as an *information bottleneck*, which prevents the model from co-adaptation of low-level colours. When deploying for downstream tasks, images of full RGB colours are taken as input directly (no jittering). *Second*, it can also be treated as data augmentation that potentially improves the model robustness under challenging cases, for instance, illumination changes.

3.3 Restricted Attention

In the previous work [40], *full attention* has been used for computing the affinity matrix, *i.e.* all pairs of pixels in target and reference frames are correlated (Figure 3 (b)). However, the memory and computational consumption tends to grow quadratically with the spatial footprint of the feature maps, thus limiting the resolution. In fact, videos are full of regularities, *i.e.* the appearances in the video clip tend to change smoothly both in spatial and temporal axes. To fully exploit this property, we propose to use a restricted attention mechanism (Figure 3 (c)), which leads to dramatic decrease in computation and memory consumption, and enables to train on *high-resolution* frames.

[1] https://arxiv.org/abs/1905.00875



(a) Colour Dropout (b) Full Attention (c) Restricted Attention

Figure 3: Restricted attention and colour dropout. See text for details.

Specifically, we impose a maximum disparity of M, *i.e.* pixels in the reference frame t are searched for locally in a square patch of size $(2M+1)\times(2M+1)$, centered at the target pixel. Suppose the feature maps have a dimension of $H\times W$, the affinity volume (A) is therefore a $4D$ tensor of dimension $H\times W\times(2M+1)\times(2M+1)$. The (i,j,k,l) entry of the tensor denotes the similarity between pixel (i,j) of the target frame, and pixel $(i+k-M, j+l-M)$ of the reference frame.

$$A^{ijkl} = \frac{\exp\langle f_t^{(i+k-M)(j+l-M)}, f_{t+1}^{ij}\rangle}{\sum_p\sum_q \exp\langle f_t^{(i+q)(j+p)}, f_{t+1}^{ij}\rangle} \tag{3}$$

$$\hat{I}_{t+1} = \psi(A_{(t,t+1)}, I_t) = \sum_p\sum_q A^{ij(p+M)(q+M)} I_t \tag{4}$$

where $p,q\in[-M,M]$, $f_t=\Phi(g(I_t);\theta)$ and $f_{t+1}=\Phi(g(I_{t+1});\theta)$ refer to the feature embeddings for frame t and $t+1$ respectively. $\psi()$ denotes a soft-copy operation for reconstructing $\hat{I}_{t+1}$ by "borrowing" colours from I_t frame.

3.4 Long-term Correspondence Flow

One of the challenges on self-supervised learning of correspondence flow is how to sample the training frames; If two frames are sampled closely in the temporal axis, the objects remain unchanged in both appearance and spatial position, matching becomes a trivial task and the model will not benefit from training on them. In the contrary, if the frames are sampled with a large temporal stride, the assumption of using reconstruction as supervision may fail, due to complex object deformation, illumination change, motion blurs, and occlusions.

In this section, we propose two approaches to improve the model's robustness to tracker drifting, and gently bridge the gap of training with samples that are neither easy nor that difficult, *i.e.* scheduled sampling, and cycle consistency.

3.4.1 Scheduled Sampling

Scheduled sampling is a widely used curriculum learning strategy for sequence-to-sequence models [3], the main idea is to replace some ground truth tokens by the model's prediction, therefore improve the robustness and bridge the gap between train and inference stage.

In our case, for n frames in a video sequence, a shared embedding network is used to get feature embeddings ($f_i=\Phi(g(I_i);\theta)$ where $i=1,\ldots,n$), the reconstruction is therefore formalized as a recursive process:

$$\hat{I}_n = \begin{cases} \psi(A_{(n-1,n)}, I_{n-1}) & (1) \\ \psi(A_{(n-1,n)}, \hat{I}_{n-1}) & (2) \end{cases}$$

while reconstructing the nth frame ($\hat{I}_n$), the model may have access to the previous frame as either groundtruth (I_{n-1}) or model prediction ($\hat{I}_{n-1}$). During training, the probability of using ground truth frames starts from a higher value (0.9) in early training stage, and is uniformly annealed to a probability of 0.6. Note that, as the model is recursive, the scheduled sampling forces the model to recover from error states and to be robust to drifting.

3.4.2 Cycle Consistency

Following the scheduled sampling, we also explicitly adopt a cycle consistency for training correspondence flow. Unlike [43], we do not use the cycle consistency as the dominating supervision signal, instead, it is treated as another regularizer for combating drifting. During training, we apply our model n frames forward and backward to the current frame.

3.5 Training Objectives

Similar to [40], we pose frame reconstruction as a classification problem, the colour for each pixel is quantized into 16 classes with K-means clustering in the *Lab* space. The objective function is defined as :

$$L = \alpha_1 \cdot \sum_{i=1}^{n} \mathcal{L}_1(I_i, \hat{I}_i) + \alpha_2 \cdot \sum_{j=n}^{1} \mathcal{L}_2(I_j, \hat{I}_j) \tag{5}$$

where $\mathcal{L}_1, \mathcal{L}_2$ refer to the pixel-wise cross entropy between groundtruth and reconstructed frames in the *forward* and *backward* paths, the loss weights for both paths are set as $\alpha_1 = 1.0$, $\alpha_2 = 0.1$ respectively, *i.e.* *forward* path is weighted more than *backward* path.

4 Experiments and Analysis

In the following sections, we start with the training details, followed by ablation studies, *e.g.* colour dropout, restricted attention, scheduled sampling and cycle consistency.

Training Details In this paper, we train CNNs in a completely self-supervised manner on Kinetics [23], meaning we do *not* use any information other than video sequences, and *not* finetune for any target task. As pre-processing, we decode the videos with a frame rate of 6fps, and resize all frames to $256 \times 256 \times 3$. In all of our experiments, we use a variant of ResNet-18 as a feature encoder (we refer the reader to our Arxiv paper for more details[2]). which ends up with feature embeddings with spatial resolution of $1/4$ the original image. The max disparity M in the restricted attention is set to be 6 (as described in Section 3.3). The temporal length n is set to 3 in our case, so when considering the forward-backward cycles, the sequence length is actually 5 frames. We train our model end-to-end using a batch size of 8 for 1M iterations with an Adam optimizer. The initial learning rate is set to $2e^{-4}$, and halved on 0.4, 0.6 and 0.8M iterations.

Evaluation Metrics In this paper, we report results on two public benchmarks: video segmentation, and pose keypoint propagation. For both tasks, a ground truth annotation is given for the first frame, and the objective is to propagate the mask to subsequent frames. In video segmentation, we benchmark our model on DAVIS-2017 [33], two standard metrics are used, *i.e.* region overlapping ($\mathcal{J}$) and contour accuracy ($\mathcal{F}$). For pose keypoint tracking, we evaluate our model on JHMDB dataset and report two different PCK metrics. The first

[2] https://arxiv.org/abs/1905.00875

($\text{PCK}_{instance}$) considers a keypoint to be correct if the Normalized Euclidean Distance between that keypoint and the ground truth is smaller than a threshold α. The second (PCK_{max}) accepts a keypoint if it is located within $\alpha \cdot \max(w,h)$ pixels of the ground truth, where w and h are the width and height of the instance bounding box.

4.1 Video Segmentation on DAVIS-2017

4.1.1 Ablation Studies

To examine the effects of different components, we conduct a series of ablation studies by removing one component at a time. All models are trained from scratch on Kinetics, and evaluated on the video segmentation task (DAVIS-2017) without finetuning.

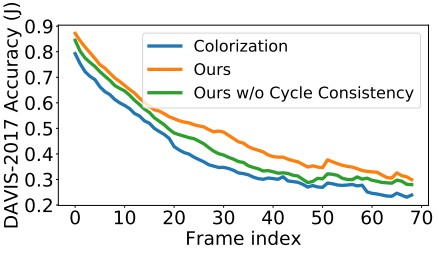


Figure 4: Model comparison on the problem of tracker drifting. The proposed model with cycle consistency has shown to be most robust as masks propagate.

Method	$\mathcal{J}$(Mean)	$\mathcal{F}$(Mean)
Ours (Full Model)	47.7	51.3
Ours w/o Colour Dropout	40.5	39.5
Ours w/o Restricted Attention	40.8	39.7
Ours w/o Scheduled Sampling	40.2	39.2
Ours w/o Cycle Consistency	41.0	40.4

Table 1: Ablation Studies on DAVIS-2017. $\mathcal{J}$: region overlapping, $\mathcal{F}$: contour accuracy respectively.

Colour Dropout Instead of taking full-colour input, we follow Vondrick *et al*. [40], and convert all frames into grayscale for inputs. As shown in Table 1, both metrics drop significantly, *i.e*. 7.2% in $\mathcal{J}$ and 11.8% in $\mathcal{F}$. This demonstrates the importance of bridging the discrepancy between training and testing on utilizing *full-RGB* colour information.

Restricted Attention When computing the affinity matrix with full attention, the model makes use of about 9.2G of GPU memory to process a single 480p image, it is therefore impossible to train on *high-resolution* frames with large batch size on standard GPUs (12-24GB). In comparison, our model with restricted attention only takes 1.4G GPU memory for the same image. As Table 1 shows, performance dropped by 6.9% and 11.6% on $\mathcal{J}$ and $\mathcal{F}$ before or after using restricted attention. This decrease confirms our assumption about spatial coherence in videos, leading both a decrease of memory consumption, and an effective regularizer that avoids the model matching correspondences very far away, for instance matching repeated patterns along the entire image.

Scheduled Sampling When not using the scheduled sampling, *i.e*. all frames used for copying are groundtruth during training, the performance dropped significantly from 47.7 to 40.2 in $\mathcal{J}$, 51.3 to 39.2 in $\mathcal{F}$, suggesting that the scheduled sampling indeed improves the model robustness under challenging scenarios, *e.g*. illumination change.

Cycle Consistency Lastly, we evalute the effectiveness of forward-backward consistency. As seen in Figure 4, while both models start off with high accuracy early in a video sequence, the model with cycle-consistency maintains a higher performance in later stages of video sequences, indicating a less severe drifting problem. This can also be reflected in the quantitative analysis, where cycle consistency enables a performance boost by 7.5% in $\mathcal{J}$ and 12.1% in $\mathcal{F}$.

4.1.2 Comparison with State-of-the-art

In Table 2, we show comparisons with previous approaches on the DAVIS-2017 video segmentation benchmark. Three phenomena can be observed: *First*, Our model clearly dominates all the self-supervised method, surpassing both video colourization (49.5 vs. 34.0 on $\mathcal{J}\&\mathcal{F}$) and CycleTime (49.5 vs. 40.7 on $\mathcal{J}\&\mathcal{F}$). *Second*, our model outperforms the optical flow method significantly, suggesting the colour dropout and scheduled sampling has improved the model's robustness under scenarios that optical flow is deemed to fail, *e.g.* large illumination changes. *Third*, our model trained with self-supervised learning can even approach the results of some supervised methods, for instance, despite of only using a ResNet18 as feature encoder, the results are comparable with the ResNet50 pretrained on ImageNet (49.5 vs. 49.7 on $\mathcal{J}\&\mathcal{F}$ (*Mean*)). We conjecture this is due to the fact that the model pretrained with ImageNet has only encoded high-level semantics, while not optimized for dense correspondence matching.

Method	Supervised	Dataset	$\mathcal{J}\&\mathcal{F}$(Mean)	$\mathcal{J}$(Mean)	$\mathcal{J}$(Recall)	$\mathcal{F}$(Mean)	$\mathcal{F}$(Recall)
Identity	✗	-	22.9	22.1	15.9	23.6	11.7
Optical Flow (FlowNet2) [15]	✗	-	26.0	26.7	-	25.2	-
SIFT Flow [28]	✗	-	34.0	33.0	-	35.0	-
Transitive Inv. [42]	✗	-	29.4	32.0	-	26.8	-
DeepCluster [50]	✗	YFCC100M	35.4	37.5	-	33.2	-
Video Colorization [40]	✗	Kinetics	34.0	34.6	34.1	32.7	26.8
CycleTime (ResNet-50) [43]	✗	VLOG	40.7	41.9	40.9	39.4	33.6
Ours (Full Model ResNet-18)	✗	Kinetics [23]	**49.5**	**47.7**	**53.2**	**51.3**	**56.5**
Ours (Full Model ResNet-18)	✗	OxUvA [39]	**50.3**	**48.4**	**53.2**	**52.2**	**56.0**
ImageNet (ResNet-50) [13]	✓	ImageNet	49.7	50.3	-	49.0	-
SiamMask [41]	✓	YouTube-VOS	53.1	51.1	60.5	55.0	64.3
OSVOS[6]	✓	DAVIS	60.3	56.6	63.8	63.9	73.8

Table 2: Video segmentation results on DAVIS-2017 dataset. Higher values are better.

4.1.3 Accuracy by Attributes

In Figure 5, we show DAVIS-2017 testing accuracy grouped into different categories. The proposed method has shown to outperform the previous methods in all situations, suggesting that our model has learned better feature embeddings for robust correspondence flow.

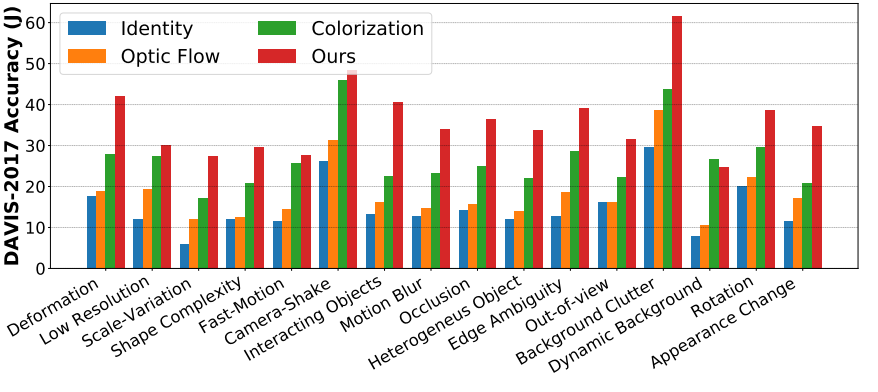


Figure 5: Accuracy by attributes.

4.1.4 Qualitative Results

As shown in Figure 1 and Figure 6, we provide the qualitative prediction from our model. The segmentation mask can be propagated through sequences even when facing large scale variation from camera motion, and object deformations.

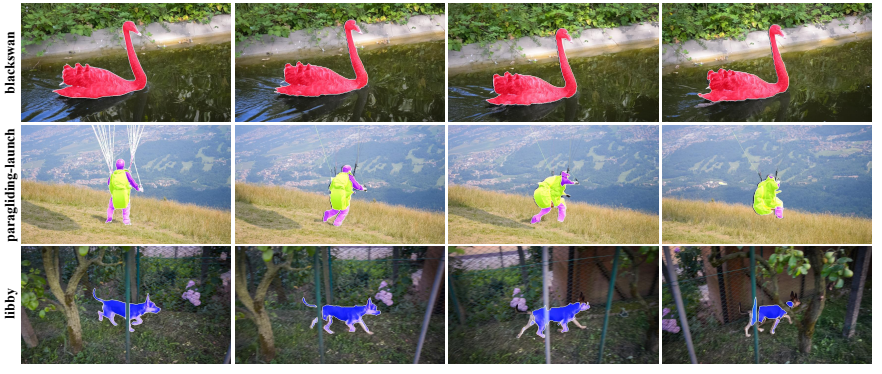


Figure 6: Qualitative results on DAVIS-2017.

4.1.5 Probing Upper Bound of Self-supervised Learning

Despite the superior performance on video segmentation, we notice that training models on Kinetics is not ideal, as it is a human-centric video dataset. However, most of the classes in DAVIS are not covered by Kinetics, *e.g.* cars, animals.

To shed light on the potential of self-supervised learning on the task of correspondence flow, we probe the upper bound by training on more diverse video data. We randomly pick 8 DAVIS classes and download 50 additional videos from YouTube, and further train the model by varying the amount of additional data. Note that, we only download videos by the class labels, and *no* segmentation annotations are provided during further self-supervised training.

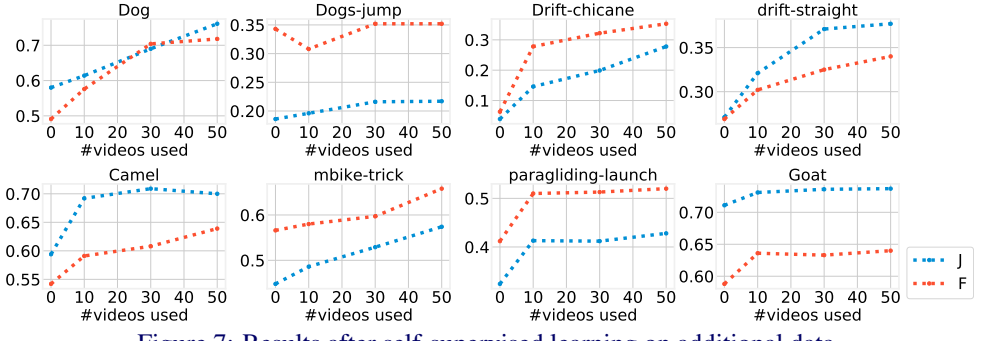


Figure 7: Results after self-supervised learning on additional data.

As shown in Figure 7 and Table 3, two phenomena can be observed: *First*, as the number of additional training videos increases (Figure 7), all sequences have seen a performance boost on both metrics $(\mathcal{J}, \mathcal{F})$. *Second*, the segmentation on some of the classes are even comparable or surpassing the supervised learning, *e.g.* Drift-c, Camel, Paragliding. We conjecture this is due to the fact that, there are only very limited manual annotations for these

classes for supervised learining, *e.g.* Paragliding. However, with self-supervised learning, we only require raw video data which is almost infinite.

Method	Dataset	Dog	Dog-j	Drift-c	Drift-s	Camel	Mbike	Paragliding	Goat
Self-supervised ($\mathcal{J}$)	Kinetics	58.0	18.6	3.9	27.1	59.4	44.8	32.4	71.1
Self-supervised ($\mathcal{J}$)	Additional	76.1	21.7	27.8	37.7	70.0	57.4	42.8	73.7
Supervised ($\mathcal{J}$) [51]	COCO+DAVIS	87.7	38.8	4.9	66.4	88.4	72.5	38.0	80.4
Self-supervised ($\mathcal{F}$)	Kinetics	49.1	34.3	6.3	26.9	54.2	56.6	41.2	58.8
Self-supervised ($\mathcal{F}$)	Additional	71.8	35.2	35.3	34.0	63.9	65.8	52.0	64.0
Supervised ($\mathcal{F}$) [51]	COCO+DAVIS	84.6	45.2	8.4	57.7	92.2	76.7	58.1	74.7

Table 3: Quantitative comparison before and after training on additional videos.

4.2 Keypoint Tracking on JHMDB

As shown in Table 4, our approach exceeds the previous methods [40] by an average of 11.3% in $\text{PCK}_{instance}$. Also, we achieve better performance in the more strict PCK@.1 metric when compared to the recent work [43]. Interestingly, when taking the benefit of the almost infinite amount of video data, self-supervised learning methods (CycleTime and Ours) achieve comparable or even outperforms model trained with the supervised learning[13, 38].

Method	Supervised	Dataset	$\text{PCK}_{instance}$		PCK_{max}	
			@.1	@.2	@.1	@.2
SIFT Flow[28]	✗	-	49.0	68.6	-	-
Video Colorization [40]	✗	Kinetics	45.2	69.6	-	-
CycleTime (ResNet-50) [43]	✗	VLOG	57.7	78.5	-	-
Ours (Full Model ResNet-18)	✗	Kinetics	**58.5**	**78.8**	**71.9**	**88.3**
ImageNet (ResNet-50) [13]	✓	ImageNet	58.4	78.4	-	-
Fully Supervised [38]	✓	JHMDB	-	-	68.7	81.6

Table 4: Keypoint tracking on JHMDB dataset (validation split 1). Higher values are better.

5 Conclusion

The paper aims to explore the self-supervised learning for pixel-level correspondence matching in videos. We proposed a simple *information bottleneck* that enables the model to be trained on standard RGB images, and nicely close the gap between training and testing. To alleviate the challenge from model drifting, we formulate the model in a recursive manner, trained with scheduled sampling and forward-backward cycle consistency. We demonstrate state-of-the-art performance on video segmentation and keypoint tracking. To further shed light on the potential of self-supervised learning on correspondence flow, we probe the upper bound by training on additional and more diverse video datasets, and show that self-supervised learning for correspondence flow is far from being saturated. As future work, potential extensions can be: *First*, explore better approaches for overcoming tracker drifting, *e.g.* use explicit memory modules for long-term correspondence flow. *Second*, define robust loss functions that can better handle complex object deformation and occlusion, *e.g.* predicting visibility mask or apply losses at feature level [32]. *Third*, instead of doing quantization with Kmeans clustering, train the quantization process to be more semantically meaningful.

Acknowledgment

Financial support for this project is provided by EPSRC Seebibyte Grant EP/M013774/1.

References

[1] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *Proc. ICCV*, 2015.

[2] C. Bailer, K. Varanasi, and D. Stricker. Cnn-based patch matching for optical flow with thresholded hinge embedding loss. In *Proc. CVPR*, 2017.

[3] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *NIPS*, 2015.

[4] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *Proc. ECCV*, 2004.

[5] T. Brox, C. Bregler, and J. Malik. Large displacement optical flow. In *Proc. CVPR*, 2009.

[6] S. Caelles, K.K Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *Proc. CVPR*, 2017.

[7] E.L. Denton and V. Birodkar. Unsupervised learning of disentangled representations from video. In *NIPS*, 2017.

[8] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proc. ICCV*, 2015.

[9] B. Fernando, H. Bilen, E. Gavves, and S. Gould. Self-supervised video representation learning with odd-one-out networks. In *CVPR*, 2017.

[10] C. Gan, B. Gong, K. Liu, H. Su, and L. J. Guibas. Geometry guided convolutional neural networks for self-supervised video representation learning. In *Proc. CVPR*, 2018.

[11] K. Han, R.S. Rezende, B. Ham, K.Y.K. Wong, M. Cho, C. Schmid, and J. Ponce. Scnet: Learning semantic correspondence. In *Proc. ICCV*, 2017.

[12] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016.

[14] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17: 185–203, 1981.

[15] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proc. CVPR*, 2017.

[16] P. Isola, D. Zoran, D. Krishnan, and E.H. Adelson. Learning visual groups from co-occurrences in space and time. In *Proc. ICLR*, 2015.

[17] T. Jakab, A. Gupta, H. Bilen, and A. Vedaldi. Conditionalimagegenerationforlearning the structure of visual objects. In *NIPS*, 2018.

[18] D. Jayaraman and K. Grauman. Learning image representations tied to ego-motion. In *Proc. ICCV*, 2015.

[19] D. Jayaraman and K. Grauman. Slow and steady feature analysis: higher order temporal coherence in video. In *Proc. CVPR*, 2016.

[20] X. Jia, B. De Brabandere, and T. Tuytelaars. Dynamic filter networks. In *NIPS*, 2016.

[21] X. Jia, R. Ranftl, and V. Koltun. Accurate optical flow via direct cost volume processing. In *Proc. CVPR*, 2017.

[22] L. Jing and Y. Tian. Self-supervised spatiotemporal feature learning by video geometric transformations. *arXiv preprint arXiv:1811.11387*, 2018.

[23] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, Viola F., T. Green, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. 2017.

[24] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proc. CVPR*, 2017.

[25] D. Kim, D. Cho, and I. S. Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *AAAI*, 2018.

[26] S. Kim, D. Min, B. Ham, S. Jeon, S. Lin, and K. Sohn. Fcss: Fully convolutional self-similarity for dense semantic correspondence. In *Proc. CVPR*, 2017.

[27] H.Y. Lee, J.B. Huang, M. Singh, and M.H. Yang. Unsupervised representation learning by sorting sequences. In *Proc. ICCV*, 2017.

[28] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 2011.

[29] S. Meister, J. Hur, and S. Roth. UnFlow: Unsupervised learning of optical flow with a bidirectional census loss. In *AAAI*, 2018.

[30] I. Misra, C. L. Zitnick, and M. Hebert. Shuffle and learn: Unsupervised learning using temporal order verification. In *ECCV*, 2016.

[31] D. Novotny, D. Larlus, and A. Vedaldi. Anchornet: A weakly supervised network to learn geometry-sensitive features for semantic matching. In *Proc. CVPR*, 2017.

[32] A.v.d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[33] J. Pont-Tuset, Perazzi. F., S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017.

[34] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *Proc. CVPR*, 2015.

[35] I. Rocco, R. Arandjelovic, and J. Sivic. Convolutional neural network architecture for geometric matching. In *Proc. CVPR*, 2017.

[36] I. Rocco, R. Arandjelovic, and J. Sivic. End-to-end weakly-supervised semantic alignment. In *Proc. CVPR*, 2018.

[37] S.M. Smith and J.M. Brady. Asset-2: Real-time motion segmentation and shape tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):814–820, 1995.

[38] J. Song, L. Wang, L. Van Gool, and O. Hilliges. Thin-slicing network: A deep structured model for pose estimation in videos. In *Proc. CVPR*, 2017.

[39] J. Valmadre, L. Bertinetto, J. F. Henriques, Tao R., A. Vedaldi, A. Smeulders, P. H. S. Torr, and E. Gavves. Long-term tracking in the wild: A benchmark. In *ECCV*, 2018.

[40] C. Vondrick, A. Shrivastava, A. Fathi, S. Guadarrama, and K. Murphy. Tracking emerges by colorizing videos. In *Proc. ECCV*, 2018.

[41] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P.H.S Torr. Fast online object tracking and segmentation: A unifying approach. In *Proc. CVPR*, 2019.

[42] X. Wang, K. He, and A. Gupta. Transitive invariance for self-supervised visual representation learning. In *Proc. CVPR*, 2017.

[43] X. Wang, A. Jabri, and A. Efros. Learning correspondence from the cycle-consistency of time. In *Proc. CVPR*, 2019.

[44] X.L Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015.

[45] Y. Wang, Y. Yang, Z. Yang, L. Zhao, P. Wang, and W. Xu. Occlusion aware unsupervised learning of optical flow. In *Proc. CVPR*, 2018.

[46] D.L. Wei, J.J. Lim, A. Zisserman, and W.T. Freeman. Learning and using the arrow of time. In *CVPR*, 2018.

[47] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. DeepFlow: Large displacement optical flow with deep matching. In *Proc. ICCV*, pages 1385–1392, 2013.

[48] O. Wiles, A.S. Koepke, and A. Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *Proc. ECCV*, 2018.

[49] L. Wiskott and T. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. In *Neural Computation*, 2002.

[50] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *Proc. ICML*, 2016.

[51] L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos. Efficient video object segmentation via network modulation. In *Proc. CVPR*, 2018.

[52] J.J. Yu, A.W. Harley, and K.G. Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *Proc. ECCV*, 2016.

Appendix A Network Architecture

We use a modified ResNet-18[13] architecture with enlarged output feature maps size. Details of the network can be found below.

0	Input image
	Feature extractor
1	7×7 conv with stride 2 and 64 filters
2	3×3 Residual Block with stride 1 and 64 filters
3	3×3 Residual Block with stride 2 and 128 filters
4	3×3 Residual Block with stride 1 and 256 filters
5	3×3 Residual Block with stride 1 and 256 filters

Table 5: Network architecture. A Residual Block stands for a residually connected sequence of operations: convolution, batch normalization, rectified linear units (ReLU), convolution, batch normalization. See [13] for details.

A.1 Failure Cases

Figure 8 demonstrates some failure cases during mask propagation. Row 1 shows our tracker fails due to occlusions. As we only use the mask of the previous frame to propagate, an object is unlikely to be retrieved after it has been occluded. Similarly, it is difficult to recover an object once it goes out of the frame (Row 2). Lastly, if the object is under complex deformation, it is likely to incur the model drifting.

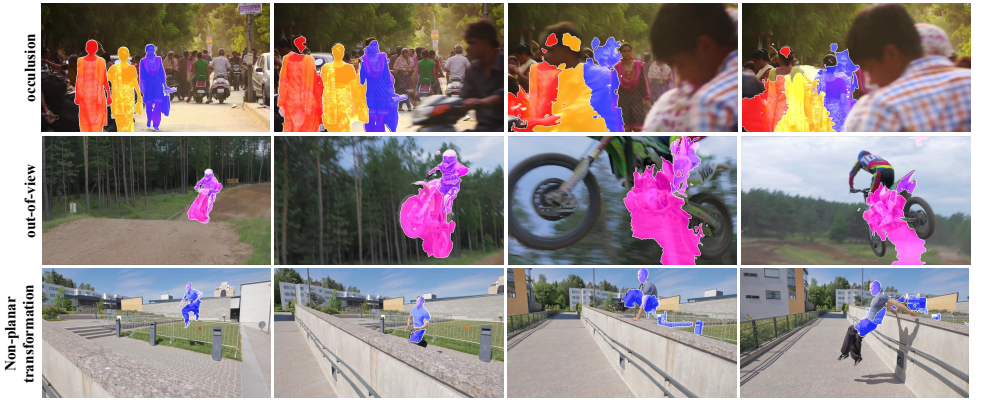


Figure 8: Common failed cases, including occulusion, out-of-view and complex transformation.