Learning Fine-Grained Features for Pixel-wise Video Correspondences

Rui Li Shenglong Zhou Dong Liu
University of Science and Technology of China, Hefei, China
{liruid,slzhou96}@mail.ustc.edu.cn, dongeliu@ustc.edu.cn

https://github.com/qianduoduolr/FGVC

Abstract

Video analysis tasks rely heavily on identifying the pixels from different frames that correspond to the same visual target. To tackle this problem, recent studies have advocated feature learning methods that aim to learn distinctive representations to match the pixels, especially in a self-supervised fashion. Unfortunately, these methods have difficulties for tiny or even single-pixel visual targets. Pixel-wise video correspondences were traditionally related to optical flows, which however lead to deterministic correspondences and lack robustness on real-world videos. We address the problem of learning features for establishing pixel-wise correspondences. Motivated by optical flows as well as the self-supervised feature learning, we propose to use not only labeled synthetic videos but also unlabeled real-world videos for learning fine-grained representations in a holistic framework. We adopt an adversarial learning scheme to enhance the generalization ability of the learned features. Moreover, we design a coarse-to-fine framework to pursue high computational efficiency. Our experimental results on a series of correspondence-based tasks demonstrate that the proposed method outperforms state-of-the-art rivals in both accuracy and efficiency.

1. Introduction

One of the most fundamental problems in computer vision is learning visual correspondences across space and time, which has many applications such as 3D reconstruction, physical understanding, and dynamic object modeling. Due to the factors such as viewpoint change, distractors, and deformations, this task is extremely challenging and can be roughly divided into three categories according to the granularity: the first one is object-wise correspondences that exist between coarsely localized bounding boxes [36, 39] along the video; the second one is group-wise correspondences, indicating the mapping at group-level, and usually applied to downstream tasks like video object segmentation [5, 25]; the third one is pixel-wise correspondences, which describe the pixel-level relation between video frames with the finest granularity.

Figure 1: We illustrate video correspondences with different granularities, including object-wise, group-wise, and pixel-wise. In this paper, we concentrate on learning fine-grained features to address the pixel-wise video correspondences.

Learning dense representations from videos is one approach to finding the correspondences. Researchers have been exploring different self-supervised methods for learning generalizable representations using unlabeled videos collected in the real world [16, 20, 23, 24, 41, 45]. For example, Wang et al. [41] propose using an object-level cycle-consistency across time (i.e., forward-backward tracking) as a supervision signal. Jabri et al. [16] further enhance it by combining cycles of time with the similarities between path-level representations. Inspired by contrastive learning, Xu et al. [45] try to learn spatial and temporal representation through a frame-wise contrastive loss, while Li et al. [23] propose a spatial-then-temporal pretext task to learn better spatiotemporal features. Despite obtaining promising outcomes, current research has predominantly emphasized performing object-level or patch-level similarity learning, making it difficult to accurately recognize pixel-wise differences with the learned features. As a result, there is an in-

This work was supported by the Natural Science Foundation of China under Grants 62022075 and 62036005, and by the Fundamental Research Funds for the Central Universities under Grant WK3490000006. This work was also supported by the advanced computing resources provided by the Supercomputing Center of USTC. *(Corresponding author: Dong Liu.)*

creasing necessity for learning fine-grained representations in order to tackle this problem.

At the same time, there is another line of work approaching video correspondences by deterministically predicting the displacement of each pixel, which is known as optical flow estimation. Early studies used optimization methods to estimate the motion between two frames [4]. In recent years, approaches use synthetic data with supervised learning for flow estimation [8, 26], using a coarse-to-fine pyramid framework to improve the accuracy [33]. RAFT [34] further devises an iterative optimization algorithm to come up with the result of high-resolution flow fields through iterative updates, which show a superior ability to find fine-grained correspondences. However, in real scenarios, there are often appearance variants, illumination changes, and deformations between video frames, which leads to the lack of robustness on real-world videos for the optical flow model supervised by labeled synthetic videos.

In this paper, we explore how to learn fine-grained representations to meet the needs of pixel-wise video correspondences. To this end, we first investigate how to leverage synthetic data for fine-grained feature learning. Specifically, given a query pixel, the supervision in synthetic videos only supplies the one-to-one mapping, i.e., a motion vector, representing the deterministic correspondence of the pixel to another pixel in the next frame. Nevertheless, the pixel-wise features evolve slowly over space and time, which indicates a soft distribution of the correspondences. We find directly utilizing the synthetic supervision as hard labels results in inferior representations, and the learned features are unable to recognize the pixel-wise differences across different spatial locations and periods of time. To address the issue, we propose to use an external pre-trained 2D encoder to derive soft supervision for optimization based on the flow.

Furthermore, we incorporate self-supervised feature learning on unlabeled real data in the overall training to alleviate the generalization issues in real scenes, which consists of two carefully designed components. Firstly, inspired by the temporal consistency assumption [3], we learn temporal persistent features via self-supervised reconstructive learning, where each query pixel can be reconstructed by leveraging the information in adjacent frames. Besides, given the synthetic and real data, we perform adversarial training by introducing Gradient Reverse Layer [9] with a discriminator for the learned correspondences. We observe such designs can further enhance learned fine-grained features.

Though already getting impressive results, we find it takes more time to get the results of the dense matching between fine-grained features. Thus, we make another step to devise a coarse-to-fine framework to address the problem. We put the complex feature matching on the coarse-grained feature map and then get the fine-grained results through a learnable up-sampling layer. As a result, we achieve a good balance of performance and efficiency. In summary, the main contribution of this work lies in:

- We address the problem of establishing pixel-wise video correspondences via a feature learning approach.
- We propose an effective method of learning fine-grained features from both synthetic and unlabeled videos, followed by a carefully designed framework to address the issue of efficiency.
- We validate our method in a series of correspondence-based tasks. Experiment results indicate consistent improvement over state-of-the-art methods.

2. Related work

Representation learning for video correspondences. Recent researches center around learning dense representations without labels in a self-supervised way for video correspondences, which occurs in two distinct directions: reconstruct-based [20, 21, 23, 24, 38, 40] and cycle-consistency-based techniques [16, 41, 49]. In the works of the first type, the query pixel is reconstructed from the adjacent frames based on the temporal consistency assumption, while the works of the second type execute forward-backward tracking to reduce cycle inconsistency. Furthermore, VFS [45] proposes to learn representations through frame-wise contrastive loss. SFC [13] proposes a two-stream architecture to learn semantic and fine-grained features through two different models. Despite the progress made in learning representations for video correspondences, accurately recognizing pixel-wise distinctions over space and time remains challenging.

Optical flow estimation for video correspondences. Recently, the classic optical flow estimation problem of predicting per-pixel motion between two frames has been explored using synthetic graphics data for supervised training [8, 26]. FlowNet [8] was one of the first deep learning methods to tackle end-to-end optical flow learning. This research inspired a multitude of other methods, such as FlowNet2.0 [15], DCFlow [44], SpyNet [30], PWC-Net [33], and LiteFlowNet3 [14]. Most of these methods employ cost volumes for finding pixel matching. RAFT [34] stands out from the rest due to its multi-scale correlation volumes and iterative flow refinements, whilst achieving superior performance, and is also a precursor to many successive works. However, learning a deterministic model with synthetic computer graphics data limits generalization ability and robustness on real videos.

Unsupervised domain adaptation with self-supervised learning. Recently, there has been a surge in approaches to reduce the distribution discrepancy between real and synthetic data by leveraging unsupervised domain adaptation [9, 19, 28, 35, 42, 48]. An effective way to

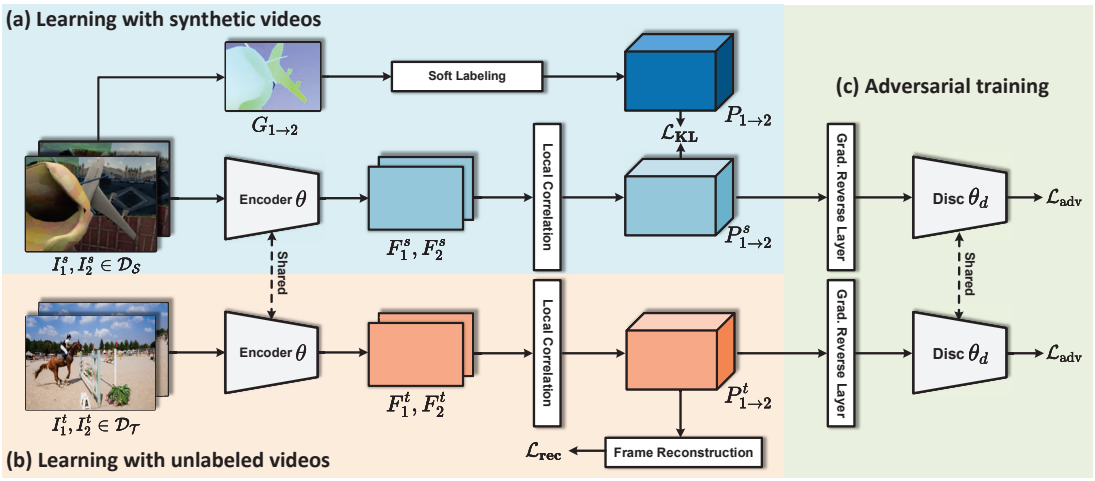

Figure 2: **Overview of the framework for fine-grained feature learning.** We facilitate fine-grained feature learning by integrating self-supervised learning on unlabeled videos into supervised learning with labeled synthetic videos. For learning with synthetic data $\mathcal{D}_\mathcal{S}$, we devise a soft labeling module to convert the hard labels indicated by the motion vectors to soft labels. To learn more generalizable features, we leverage free supervision from unlabeled videos $\mathcal{D}_\mathcal{T}$ with the objective of frame reconstruction. Furthermore, an adversarial loss (along with Gradient Reverse Layer and a discriminator) is proposed to encourage domain invariant representations. The whole framework is jointly optimized in an end-to-end manner with the proposed objective functions.

realize it is through adversarial training. Recent studies execute adversarial training by learning a domain classifier (the discriminator) to distinguish the learned features from different distributions, utilizing adversarial loss to increase domain confusion. Meanwhile, self-supervised learning obtains impressive performance and shows good generalizing ability by designing different pretext tasks [11, 41, 43, 45] with unlabeled data, which motivates us to combine supervised learning and self-supervised learning to encourage consistent representations for both domains as well as improved results for downstream tasks.

3. Approach

We address the problem of estimating the pixel-wise correspondences between a pair of video frames, which can be realized by learning fine-grained features for matching. Our goal is to learn a fine-grained feature space ϕ with the encoder θ by designing different learning objective functions for the probabilistic mapping.

Probabilistic mapping. Given a pair of video frames $I_1, I_2 \in \mathbb{R}^{h\times w\times 3}$, for all pixels in I_1, we aim to predict the probabilistic mapping $P_{1\to2} \in \mathbb{R}^{H\times W\times(2r+1)^2}$, and $P_{1\to2}(\cdot|i) \in \mathbb{R}^{(2r+1)^2}$ gives the probability that i is mapped to j in frame I_2 within a limited range r, considering the nature of temporal coherence in the video. The $i, j \in \mathbb{R}^2$ indicate the 2D pixel location. $P_{1\to2}(\cdot|i) \in \mathbb{R}^{(2r+1)^2}$ thus encodes the entire discrete conditional probability distribution of where i is mapped in frame I_2. The probabilistic mapping can be achieved by calculating the feature similarities using the learned fine-grained features. More specifically, we first extract the dense features $F_1, F_2 \in \mathbb{R}^{H\times W\times C}$. Then the discrete probabilistic map can be obtained by computing the local correlation w.r.t. each key j in I_2 within a local window for each query i,

$$P_{1\to2}(j|i) = \frac{\exp\left(F_1(i)\cdot F_2(j)/\tau\right)}{\sum_n \exp\left(F_1(i)\cdot F_2(n)/\tau\right)}, i \in \{1,.,HW\}, j, n \in \mathcal{K}(i), \tag{1}$$

where $\mathcal{K}(i)$ is the index set in the local window with a limited range of r centered at i, and τ is the temperature. The result of the probabilistic mapping is further post-processed and directly applied to various downstream tasks.

3.1. Fine-Grained Feature Learning

We design the learning objective functions for probabilistic mapping with both synthetic and real-world videos. The overview of the framework is shown in Figure 2.

Learning with synthetic videos. Labeled synthetic videos are often used as supervision for learning the optical flows in recent studies [8, 34]. Given a pair of rendered video frames I_1^s, I_2^s, the synthetic data $\mathcal{D}_\mathcal{S}$ provides pixel-wise motion vector, i.e., optical flow $G_{1\to2}$. A valid question then emerges as how to learn features using such deterministic correspondences? Indeed, we believe the deterministic correspondences are hard to obtain for real-world videos, thus we argue the necessity to convert them into soft (probabilistic) ones. We devise the following variants:

(i) *Dirac distribution*: As shown in Figure 3 (a), we can directly convert the motion vector to a dirac distribution $\delta(\cdot|i)$ in order to describe the ground truth mapping. Then, the learning objective function defined as Kullback-Leibler divergence between $P_{1\to2}^s(\cdot|i)$ and $\delta(\cdot|i)$. The $P_{1->2}^s$ stands

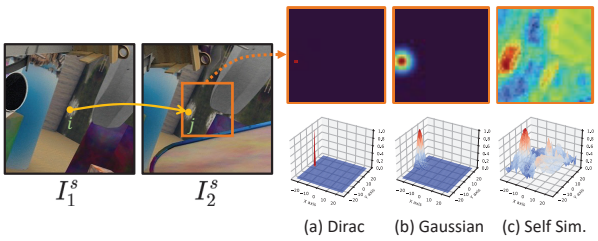

Figure 3: **Illustration of the probabilistic map.** For i in I_1^s, we visualize the probability value of the mapping from i to the locations in I_2^s within a limited range.

for the correlation calculated by Eq. (1) with synthetic data:

$$\mathcal{L}_{\text{KL-v1}} = \sum_i D_{KL}(\delta(\cdot|i) \parallel P^s_{1\to 2}(\cdot|i)). \tag{2}$$

(ii) *Gaussian distribution*: However, the feature of the query pixel i varies smoothly over space and time, which indicates a soft distribution of the potential location in the next frame. As shown in Figure 3 (a), the dirac distribution does not provide the feature learning with any knowledge of relative probability in the background, making it hard to learn the features with the ability to tell the fine-grained differences with synthetic videos. Thus, we devise to generate the probabilistic map with gaussian distribution, which introduces a soft distribution centered at the ground truth coordinate:

$$\mathcal{N}\left(j|i,\mu,\sigma^2\right) = \frac{1}{\sqrt{2\pi}\sigma_u} e^{-\frac{\left(u_j-\mu_i^u\right)^2}{2\sigma_u^2}} \cdot \frac{1}{\sqrt{2\pi}\sigma_v} e^{-\frac{\left(v_j-\mu_i^v\right)^2}{2\sigma_v^2}}, \tag{3}$$

where $\mu = (\mu_i^u, \mu_i^v)$ represents the 2D coordinate of ground truth in the I_2^s for pixel i, and (u_j, v_j) indicates the 2D coordinate of j. Then the loss can be devised as:

$$\mathcal{L}_{\text{KL-v2}} = \sum_i D_{KL}(\mathcal{N}(\cdot|i,\mu,\sigma^2) \parallel P^s_{1\to 2}(\cdot|i)). \tag{4}$$

(iii) *Soft-labeling*: The gaussian prior only considers the euclidean distance of coordinates, which shows limited capability of modeling the complicated distributions. We believe the synthetic supervision already provides valuable temporal cues (i in I_1^s move to j in I_2^s), and the soft distribution over space can be approached by computing the self-similarities for j using the pre-trained 2D visual encoder θ_{self} which can produce spatially-discriminative features. As shown in Figure 3 (c), for the i in I_1^s that moves to j, we get the feature of the query i (denoted as $\overline{F}_q$) at the location j in $\overline{F}_2$ ($\overline{F}_2$ is the feature map of I_2^s computed by θ_{self}). Then we can compute the feature similarities $S_{1\to 2}(\cdot|i)$ between $\overline{F}_q$ and $\{\overline{F}_2(k)|k \in \mathcal{K}(i)\}$, where $\mathcal{K}(i)$ is the index set in the local window centered at i. The $S_{1\to 2}(\cdot|i)$ are further normalized (by softmax) to obtain the discrete probability distribution $P_{1\to 2}(\cdot|i)$:

$$\mathcal{L}_{\text{KL-v3}} = \sum_i D_{KL}(P_{1\to 2}(\cdot|i) \parallel P^s_{1\to 2}(\cdot|i)). \tag{5}$$

In Table 4, we find soft labeling works well when only pre-training θ_{self} on synthetic data $\mathcal{D}_\mathcal{S}$ with $\mathcal{L}_{\text{rec}}$ (Eq. (8)), and leveraging more strong 2D encoder would contribute to better performance. The comparisons between different loss functions will be discussed in the experiments, and the $\mathcal{L}_{\text{KL-v3}}$ is used as the default loss.

Learning with unlabeled videos. Meanwhile, we observe in real scenarios, there are apparent differences with synthetic videos in appearance variants, illumination changes and deformations, leading to notable changes in the distribution, where the learned features on synthetic videos show unsatisfied generalization ability. Inspired by recent studies [16, 41], we try to improve the learned fine-grained features by introducing self-supervised feature learning into the framework. As observed in the bottom of Figure 1, the pixel repetition encourages us to learn the fine-grained features by reconstructive learning, where each pixel in the I_1^t can be reconstructed by leveraging the information of I_2^t with a limited range. To achieve this, the video frames I_1^t, I_2^t are firstly projected into pixel embeddings F_1^t, F_2^t by the encoder θ. For each query i in I_1^t, we calculate the probabilistic map $P^t_{1\to 2}$ with Eq. (1). Then the query i in I_1^t can be reconstructed by a weighted sum of pixels in $\mathcal{K}(i)$:

$$I_1^{rec}(i) = \sum_{j\in\mathcal{K}(i)} P^t_{1\to 2}(j|i) I_2^t(j). \tag{6}$$

Then the reconstruction loss for self-supervised training is defined as L_1 distance between I_1^{rec} and I_1^t. Training with such self-supervision leads to temporal persistent features that generalize well in real scenarios.

However, the pixel repetition does not hold for pixels that become occluded, Thus, we exclude occluded pixels from the reconstruction loss to avoid learning incorrect features. We follow the forward-backward consistency assumption to detect the occluded pixels. which is defined in Eq. (7) as the occlusion flag $O_{1\to 2}$ to be 1 whenever the constraint is violated, and 0 otherwise:

$$O_{1\to 2}(i) = \mathbb{1}\left(\arg\max_i P^t_{2\to 1}\left(i \mid \arg\max_j P^t_{1\to 2}(j \mid i)\right) = i\right). \tag{7}$$

The loss for reconstructive learning is defined as follows:

$$\mathcal{L}_{\text{rec}} = \sum_i O_{1\to 2}(i) \cdot \left\| I_1^{rec}(i) - I_1^t(i) \right\|_1. \tag{8}$$

Adversarial training. We further improve the fine-grained features by leveraging the technique in recent works of unsupervised domain adaptation, which aims to bridge the gap caused by the domain shift between the synthetic videos $\mathcal{D}_\mathcal{S}$ and real videos $\mathcal{D}_\mathcal{T}$ via adversarial training. An effective way to approach this problem consists in introducing the network a Gradient Reversal Layer (GRL). We additionally train a discriminator θ_D to identify whether the

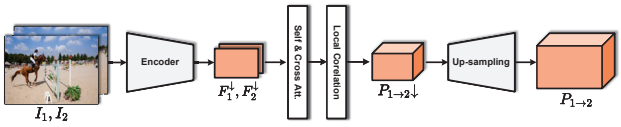

Figure 4: **Illustration of the coarse-to-fine framework for efficient probabilistic mapping.** We first obtain coarse-grained matching and then upsample it to get the fine-grained result.

probabilistic map comes from synthetic or real videos:

$$\mathcal{L}_{\text{adv}} = \mathbb{E}_{P^t \in P_{\mathcal{T}}} \left[\log\left(D(P^t)\right)\right] + \mathbb{E}_{P^s \in P_{\mathcal{S}}} \left[\log\left(1 - D(P^s)\right)\right], \tag{9}$$

and then we reverse the gradient direction during the backward pass in back-propagation when updating the parameters of the encoder θ with $\mathcal{L}_{\text{adv}}$:

$$\theta \longleftarrow \theta + \lambda \frac{\partial \mathcal{L}_{\text{adv}}}{\partial \theta}. \tag{10}$$

The GRL allows to train the discriminator and the encoder at the same time, and the adversarial training helps to learn domain invariant patterns.

Overall training objective. The overall training objective for learning fine-grained features is formulated as a multi-task loss, which is written as $\mathcal{L} = \mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{adv}}$, where we empirically treat each loss equally.

3.2. Efficient Fine-grained Probabilistic Mapping

To get the fine-grained probabilistic map, i.e., $P_{1\to 2} \in \mathbb{R}^{H\times W\times(2r+1)^2}$, we need to compute the similarities w.r.t. all key pixels in the local window centered at every query pixel, which is computationally costly. In this section, we devise a coarse-to-fine framework. The overview of the framework can be found in Figure 4, where we first compute the correlation at coarse-grained features and then upsample it to get fine-grained probabilistic maps.

Self-attention and Cross-attention Layers. More specifically, we first extract the coarse feature maps $F_1^{\downarrow}, F_2^{\downarrow} \in \mathbb{R}^{H/4\times W/4\times C}$, and we further enhance the coarse features by introducing the self-attention and cross-attention layers with positional encoding. Then, we obtain the coarse-grained probabilistic map $P_{1\to 2\downarrow} \in \mathbb{R}^{H/4\times W/4\times(2r\downarrow+1)^2}$ by Eq (1).

Up-sampling. We devise an up-sampling layer to obtain fine-grained probabilistic map $P_{1\to 2} \in \mathbb{R}^{H\times W\times(2r+1)^2}$, which can be simply done by leveraging pixel-shuffle or convolution layer with bilinear interpolation.

More details of the architecture are included in the supplementary material. We train the coarse-to-fine framework in a distillation manner on $\mathcal{D}_{\mathcal{T}}$, with the same objective function as $\mathcal{L}_{\text{KL}}$. The supervisions for $P_{1\to 2}$ are obtained by computing the probabilistic map using pre-trained encoder in Sec. 3.1. Without losing much performance, we find such a design helps to get rid of the complicated fine-grained feature matching and exhibit higher efficiency.

4. Experiments

We verify the effectiveness of our method in a series of correspondence-based tasks. We will first introduce implementation and evaluation details, and report the performance comparison with baselines. Finally, we perform detailed ablation studies for each component of our method.

4.1. Implementation Details

Backbone. We exploit the encoder θ with ResNet-18 [12]. We reduce the stride up to layer res_4 to get features at $1/2$ of the original image dimension for training. In our coarse-to-fine framework, we increase the stride of the encoder to 8 for coarse-grained feature matching.

Training details. The training is conducted on the train set of synthetic dataset FlyingThings [26] and YouTube-VOS [46] collected in real-world. The FlyingThings/YouTube-VOS contains 40k/3.5k videos for training. For both datasets, we sample pair of frames which are resized into 256×256 and converted to Lab color space with channel-wise dropout as the information bottleneck [20]. The local range r in the probabilistic map is set to 24/6 for the fine-grained features ($\frac{h}{2} \times \frac{w}{2}$ (stride=2)) or coarse-grained features ($\frac{h}{8} \times \frac{w}{8}$ (stride=8)). We first train an encoder using reconstruction loss (Eq. (8)) with a batchsize of 32 for 30 epochs on FlyingThings [26], which is further utilized as θ_{self} in soft labeling. Then the final model θ is jointly trained for 30 epochs with proposed three losses, with a batchsize of 16 for each dataset. When training with synthetic optical flow, we filter out the points that are out of the local range or occluded by forward-backward checking. The τ is set to 1. We use Adam as the optimizer, and the initial learning rate is set to 1e-3 with a cosine (half-period) learning rate schedule for each stage. We include more training details in the supplementary.

Evaluation. An important problem is how to evaluate the quality of the learned fine-grained features for pixel-wise video correspondences. Following the previous works [16, 41, 45], without any fine-tuning, we directly leverage the pre-trained encoder θ to extract features F_i, F_j with a spatial resolution of $\frac{h}{s} \times \frac{w}{s}$ (the s represents the stride of the encoder) for the pair of frames I_i, I_j, which are later used to compute the probabilistic map $P_{i\to j}$ by Eq (1). Based on the probabilistic map $P_{i\to j}$, we follow the recurrent inference strategy inference strategies of recent studies [16, 41, 45] to propagate the target points or semantical labels of the first frame, as well as previous predictions, to the current frame I_t. We evaluate the point tracking on three popular benchmarks including BADJA [2], JHMDB [18], TAP-Vid-DAVIS [7] and TAP-Vid-Kinetics [7], and the evaluation of video object segmentation is conducted on the widely used dataset DAVIS-2017 [29].

Table 1: **Quantitative results for point tracking on different datasets.** The frame per second (FPS) results of getting pixel-wise correspondences between a pair of video frames are measured on a single GTX-3090 at the resolution of 480×640. The * indicates our coarse-to-fine framework. The - indicates the unavailable result due to the unavailable pre-trained model.

Method	Backbone	Stride	FPS	BADJA	JHMDB	TAP-DAVIS	TAP-Kinetics
				PCK@0.1↑	PCK@0.1↑	$< \delta^x_{avg-p}$ ↑	$< \delta^x_{avg-p}$ ↑
TimeCycle [41]	ResNet-50	8	28	41.1	57.3	27.1	28.6
UVC [24]	ResNet-18	8	142	48.2	58.6	29.0	25.2
CRW [16]	ResNet-18	8	142	50.9	59.3	32.5	25.4
VFS [45]	ResNet-18	8	142	51.9	60.5	31.9	28.8
CLSC [31]	ResNet-18	8	142	-	61.7	-	-
SFC [13]	ResNet-18 + ResNet-50	8	27	53.8	61.9	37.2	31.1
MAST [20]	ResNet-18	4	33	55.7	62.4	42.5	33.2
LIIR [22]	ResNet-18	4	33	-	60.7	-	-
Ours*	ResNet-18	8	34	**56.8**	**64.6**	**48.0**	**43.8**
ImageNet Pre. [12]	ResNet-18	2	8	57.5	61.5	51.3	44.5
UVC [24]	ResNet-18	2	8	56.7	65.1	52.7	41.8
CRW [16]	ResNet-18	2	8	55.9	61.4	43.2	37.1
VFS [45]	ResNet-18	2	8	58.1	61.0	51.4	44.9
SFC [13]	ResNet-18	2	8	61.5	65.9	53.9	43.6
MAST [20]	ResNet-18	2	8	63.0	63.1	53.8	42.7
Ours	ResNet-18	2	8	**67.2**	**66.8**	**59.8**	**48.8**

4.2. Results for Point Tracking

We firstly make comparisons for point tracking since it requires the finest granularity of the learned features. The main comparators of our method are the works aim to learn good representations for matching, e.g., TimeCycle [41], UVC [24], CRW [16], VFS [45], SFC [13], and MAST [20]. We also include the model pre-trained on ImageNet [6] with human annotations. These works share a similar evaluation protocol as we mentioned in 4.1.

While previous works test the pixel-wise correspondences on JHMDB [18] that only provides the human keypoint annotations. We additionally include BADJA [2], TAP-Vid-DAVIS [7] and TAP-Vid-Kinetics [7]. We propagate the points of the first frame to other frames and evaluate the results using the annotations of each dataset. We notice some works are trained and evaluated with the coarse-grained features (e.g., the features at 1/8 of the original image dimension). For these methods, we use our coarse-to-fine framework that also executes coarse-grained feature matching for comparisons. Meanwhile, for better comparisons, we follow the studies in [16, 45] to further reduce the stride s of the encoder to 2, in order to get more fine-grained results for CRW, VFS, SFC and MAST. We also provide the FPS of computing the pixel-wise correspondences between two frames for feature-matching-based methods, which can be done by first taking the index of the maximum value in $P_{1\to 2}(\cdot|i)$ for each i and then applying up-sampling to get full-resolution results. More details about the evaluation are included in the supplementary material.

Results on BADJA/JHMDB. We adopt the standard PCK [47] of all visible points (not compute PCK for each video then take average) as the evaluation metric. Each point is considered correct if it is within a distance of $0.1\sqrt{A}$ from the ground truth, where A is the distance between keypoints (for JHMDB) or the area of the ground-truth segmentation mask on the frame (for BADJA). In Table 1, our method achieves 67.2%/66.8%, surpassing all state-of-the-art methods. Besides, our method with the coarse-to-fine design still makes the absolute improvements by 1.1% and 2.2% compared with MAST, and shows better efficiency compared with SFC [13] that uses the two-stream network to find the correspondences. More remarkably, our efficient framework even surpasses part of the methods using more fine-grained features for inference.

Results on TAP-Vid. TAP-Vid is a newly developed benchmark composed of long-term videos in real-world with accurate human annotations of point tracks. We test on the whole set of TAP-Vid-DAVIS and the test set of TAP-Vid-Kinetics. We adopt the setting of "first fashion" in [7], which tracks only into the future. The average position accuracy over all visible points ($< \delta^x_{avg-p}$) is adopted as the metric. The learned fine-grained features obtain 59.8%/48.8% on TAP-Vid-DAVIS/TAP-Vid-Kinetics, leading apparent improvements over state-of-the-arts by 5.9%/3.9%. Moreover, the proposed coarse-to-fine framework gets 48.0%/43.8%, surpassing MAST by 5.5%/10.6%.

Comparisons with task-specific methods. Besides, we notice there are some recent methods specifically designed for point tracking, like RAFT [34], PIPs [10], TAPNet [7], and Thin-Slicing Net [32] even trained with human annotations. Here we also present the performance comparisons with them in Table 2. It's worth noting that, to align the evaluation with these methods, except JHMDB, we first compute the $< \delta^x_{avg-p}$ or PCK for each video to obtain video-level results, and the final results are obtained by taking the average over all videos. Without any specific designs, our performance even surpasses these methods by 7.4%/7.5%/0.2% on BADJA/TAP-DAVIS/TAP-Kinetics.

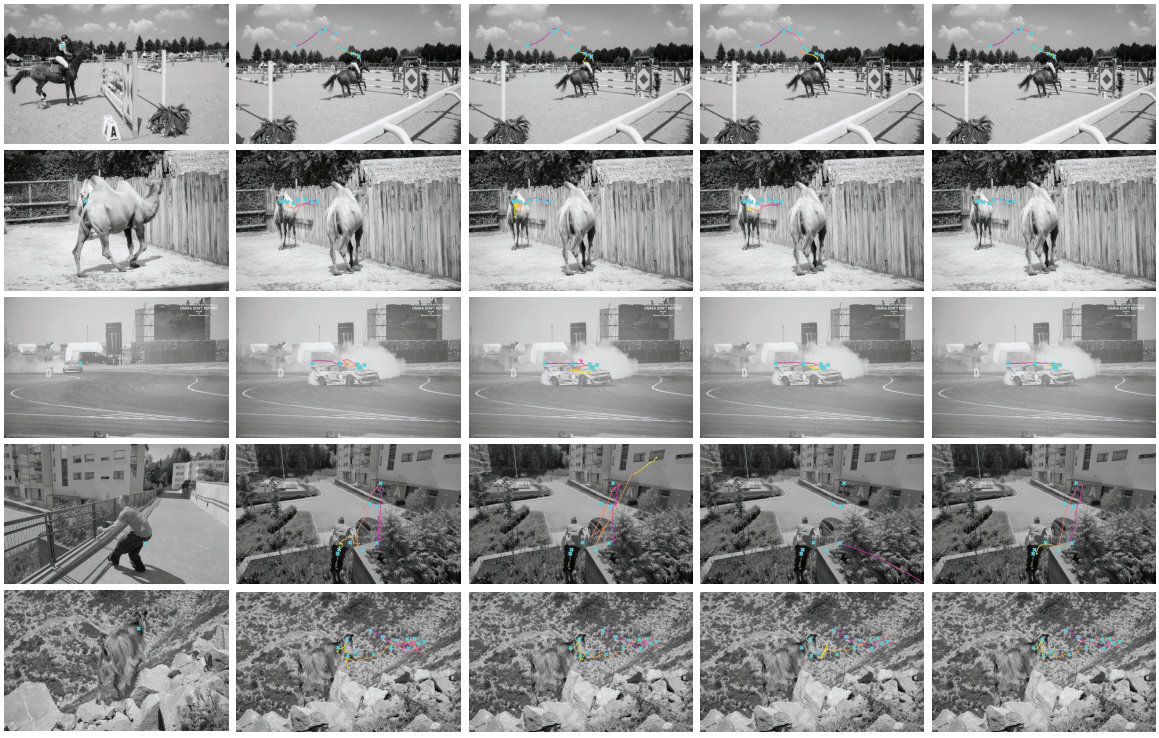

Figure 5: **Qualitative results for point tracking.** Given the target pixel in the first frame, we visualize the estimated trajectory with a pink-to-yellow colormap (pink/yellow indicates the start/end of the video clip). We visualize sparse ground-truth labels with cyan cross marks. Please refer to the video in the supplementary for better animations. **(Zoom in for best view)**

Table 2: **Comparisons with methods specifically designed for point tracking.** "‡" means we align the evaluation protocol with each method for fair comparisons.

Method	BADJA	JHMDB	TAP-DAVIS	TAP-Kinetics
	PCK@0.2 ↑	PCK@0.1 ↑	$< \delta^x_{avg}$ ↑	$< \delta^x_{avg}$ ↑
RAFT [34]	45.6	66.4	42.1	44.3
PIPs [10]	62.3	-	55.3	48.2
TAPNet [7]	-	62.3	48.6	54.4
Thin-Slicing Net [32]	-	**68.7**	-	-
Ours‡	**69.7**	66.8	**62.8**	**54.6**

Table 3: **Quantitative results for video object segmentation on DAVIS$_{17}$** [29]. "Sup." means using human annotations for training.

Method	Sup.	Backbone	DAVIS$_{17}$		
			$\mathcal{J}\&\mathcal{F}_m$ ↑	$\mathcal{J}_m$ ↑	$\mathcal{F}_m$ ↑
TimeCycle [41]		ResNet-50	40.7	41.9	39.4
UVC [24]		ResNet-18	59.5	57.7	61.3
MAST [20]		ResNet-18	65.5	63.3	67.6
CRW [16]		ResNet-18	67.6	64.8	70.2
JSTG [49]		ResNet-18	68.7	65.8	71.6
VFS [45]		ResNet-50	68.9	66.5	71.3
DUL [1]		ResNet-18	69.3	67.1	71.6
MAMP [27]		ResNet-18	69.7	68.3	71.2
CLTC [17]		ResNet-18	70.3	67.9	72.6
CLSC [31]		ResNet-18	70.5	67.4	73.6
SFC [13]		ResNet-18 + ResNet-50	71.2	68.3	74.0
LIIR [22]		ResNet-18	72.1	69.7	74.5
Ours		ResNet-18	**72.4**	**70.5**	**74.4**
OSVOS-S [25]	✓	VGG-16	68.0	64.7	71.3
FEELVOS [37]	✓	Xception-65	71.5	69.1	74.0

Figure 5 shows some visualization results of the point tracking on TAP-Vid-DAVIS. Given the target in the first frame, we visualize the estimated trajectory with a pink-to-yellow map. Compared with VFS, MAST and RAFT, our approach can output more smooth and accurate trajectories close to the sparse visualized ground truth, even facing dramatic appearance changes and deformation.

4.3. Results for Video Object Segmentation

Next, we evaluate methods with semi-supervised video object segmentation. We use the mean of region similarity $\mathcal{J}_m$, mean of contour accuracy $\mathcal{F}_m$ and their average $\mathcal{J}\&\mathcal{F}_m$ as the evaluation metrics. In Table 3, our method still leads the performance. More remarkably, our method even outperforms some task-specific fully-supervised algorithms [25,37]. Here we select several representative videos for inference, and give the visualization results in Figure 6, our method produces tight boundaries around the object areas, and obtains more fine-grained results, especially for small objects. For example, in the first column of Figure 6, the tiny arm of the human can still be segmented, which further demonstrates the advantages of learning fine-grained features for video correspondences. However, we

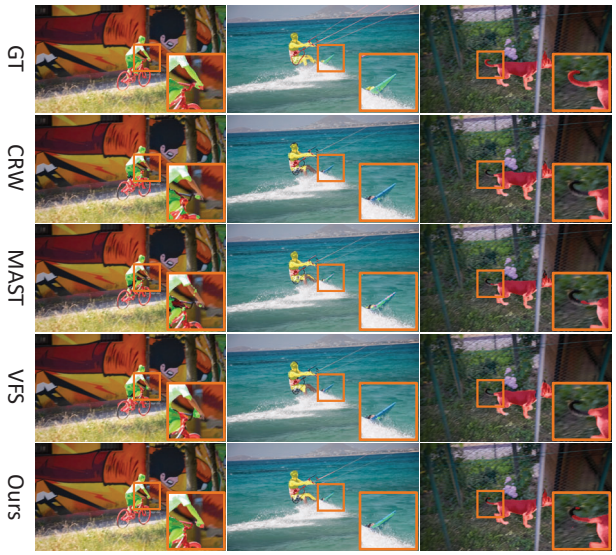

Figure 6: **Qualitative results for video object segmentation.** Given the semantic mask in the first frame, we show the propagation results on the target frame. (**Zoom in for best view**)

find that fine-grained features may hinder object-centric feature learning since it may rely more on low-level patterns (e.g., texture, color, etc), which degrades the performance in video object segmentation to some extent. We regard it as our future work.

4.4. Ablation Study

We perform ablation study with point tracking on the TAP-Vid-DAVIS [7] dataset.

Learning with synthetic videos. We study the effect of three different ways as defined in Eq. (2), Eq. (4) and Eq. (5) for learning the fine-grained features with synthetic videos. Table 4 shows the performance comparisons across the three kinds of objective functions. As indicated by the results, the loss $\mathcal{L}_{\text{KL-v1}}$ using dirac distribution to generate the labels performs badly. We think the deterministic labels were given on synthetic videos, resulting in the learned features not being robust enough on real-world videos. Besides, we find introducing the gaussian distribution in $\mathcal{L}_{\text{KL-v2}}$ results in performance degradation. The performance drops a lot when progressively increasing the variance, which may be attributed to the inability of gaussian distribution to approach the real probabilistic distribution since it is extremely complicated. Expectedly, the proposed soft labeling boosts up the performance from 42.6% to 55.8% when only pre-training the θ_{self} with Eq. (8). Moreover, we also try another 2D encoder pre-trained with the contrastive loss $\mathcal{L}_{\text{nce}}$ [11] on ImageNet [6], which has a stronger ability to capture the spatially discriminative features. The results are further improved to 57.5%, which motivates us to leverage a more powerful 2D feature extractor for obtaining the soft labels. We regard it as future work.

Table 4: **Ablation study for $\mathcal{L}_{\text{KL}}$.** The σ / θ_{self} represents the variance / encoder used in $\mathcal{L}_{\text{KL-v2}}$ / $\mathcal{L}_{\text{KL-v3}}$.

Obj. Function	Hyper-param.		TAP-Vid-DAVIS
	(σ_u, σ_v)	θ_{self}	$< \delta^x_{avg} \uparrow$
$\mathcal{L}_{\text{KL-v1}}$	-	-	42.6
$\mathcal{L}_{\text{KL-v2}}$	(1,1)	-	41.9
	(3,3)	-	33.8
	(6,6)	-	29.7
$\mathcal{L}_{\text{KL-v3}}$	-	w. $\mathcal{L}_{\text{rec}}$	55.8
	-	w. $\mathcal{L}_{\text{nce}}$ [11]	**57.5**

Table 5: **Ablation study for training objective functions.** FT: FlyingThings [26]. YTV: YouTube-VOS [46].

$\mathcal{L}_{\text{KL}}$	$\mathcal{L}_{\text{rec}}$	$\mathcal{L}_{\text{adv}}$	Training Data	TAP-Vid-DAVIS $< \delta^x_{avg} \uparrow$
✓			FT	55.8
	✓		YTV	56.4
	✓		FT + YTV	56.7
✓	✓		FT + YTV	59.2
✓	✓	✓	FT + YTV	**59.8**

Different training objective functions. We examine how different objective functions impact the overall performance, which is shown in Table 5. The $\mathcal{L}_{\text{KL}}$, $\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_{\text{adv}}$ denotes the losses defined in Eq. (5), Eq. (8) and Eq. (9). Surprisingly, we find the model trained on unlabeled real-world videos with $\mathcal{L}_{\text{rec}}$ obtains a better result, which indicates better generalization of self-supervised feature learning. By leveraging both $\mathcal{L}_{\text{KL}}$ and $\mathcal{L}_{\text{rec}}$ with synthetic and real-world videos, the performance is further improved to 59.2%. As expected, executing $\mathcal{L}_{\text{adv}}$ to address the domain mismatch improves performance by 0.6%. By fusing three losses, the performance reaches 59.8%. The results consistently indicate that incorporating self-supervised and adversarial training with unlabeled data exhibits a performance boost against the model only trained with synthetic data.

5. Conclusions

In this paper, we address pixel-wise video correspondences by learning fine-grained features. We propose to use not only labeled synthetic videos but also unlabeled real-world videos for feature learning. We first study how to take advantage of synthetic supervision for feature learning, and we find directly utilizing the motion vector results in degradation for the learned features. Thus, we propose soft labeling to address the issue. To improve the generalization, we introduce self-supervised reconstructive learning into the overall training and further enhance the features by leveraging adversarial training. Moreover, we propose a coarse-to-fine framework to alleviate the problem of computational efficiency. Extensive experiments on the downstream tasks validate the effectiveness of the proposed feature learning method and our efficient design.

References

[1] Nikita Araslanov, Simone Schaub-Meyer, and Stefan Roth. Dense unsupervised learning for video segmentation. In *NeurIPS*, pages 25308–25319, 2021.

[2] Benjamin Biggs, Thomas Roddick, Andrew Fitzgibbon, and Roberto Cipolla. Creatures great and smal: Recovering the shape and motion of animals from video. In *ACCV*, pages 3–19, 2019.

[3] Michael J Black and Padmanabhan Anandan. A framework for the robust estimation of optical flow. In *ICCV*, pages 231–236, 1993.

[4] Thomas Brox and Jitendra Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE transactions on PAMI*, 33(3):500–513, 2010.

[5] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, pages 221–230, 2017.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[7] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adrià Recasens, Lucas Smaira, Yusuf Aytar, João Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. In *NeurIPS*, 2022.

[8] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, pages 2758–2766, 2015.

[9] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, pages 1180–1189, 2015.

[10] Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *ECCV*, pages 59–75, 2022.

[11] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[13] Yingdong Hu, Renhao Wang, Kaifeng Zhang, and Yang Gao. Semantic-aware fine-grained correspondence. In *ECCV*, pages 97–115, 2022.

[14] Tak-Wai Hui and Chen Change Loy. Liteflownet3: Resolving correspondence ambiguity for more accurate optical flow estimation. In *ECCV*, pages 169–184, 2020.

[15] 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*, pages 2462–2470, 2017.

[16] Allan Jabri, Andrew Owens, and Alexei Efros. Space-time correspondence as a contrastive random walk. In *NeurIPS*, pages 19545–19560, 2020.

[17] Sangryul Jeon, Dongbo Min, Seungryong Kim, and Kwanghoon Sohn. Mining better samples for contrastive learning of temporal correspondence. In *CVPR*, pages 1034–1044, 2021.

[18] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *ICCV*, pages 3192–3199, 2013.

[19] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *CVPR*, pages 4893–4902, 2019.

[20] Zihang Lai, Erika Lu, and Weidi Xie. Mast: A memory-augmented self-supervised tracker. In *CVPR*, pages 6479–6488, 2020.

[21] Zihang Lai and Weidi Xie. Self-supervised learning for video correspondence flow. In *BMVC*, 2019.

[22] Liulei Li, Tianfei Zhou, Wenguan Wang, Lu Yang, Jianwu Li, and Yi Yang. Locality-aware inter-and intra-video reconstruction for self-supervised correspondence learning. In *CVPR*, pages 8719–8730, 2022.

[23] Rui Li and Dong Liu. Spatial-then-temporal self-supervised learning for video correspondence. In *CVPR*, pages 2279–2288, 2023.

[24] Xueting Li, Sifei Liu, Shalini De Mello, Xiaolong Wang, Jan Kautz, and Ming-Hsuan Yang. Joint-task self-supervised learning for temporal correspondence. In *NeurIPS*, pages 318–328, 2019.

[25] K-K Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. Video object segmentation without temporal information. *IEEE transactions on PAMI*, 41(6):1515–1530, 2018.

[26] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, pages 4040–4048, 2016.

[27] Bo Miao, Mohammed Bennamoun, Yongsheng Gao, and Ajmal Mian. Self-supervised video object segmentation by motion-aware mask propagation. In *ICME*, pages 1–6, 2022.

[28] Yingwei Pan, Ting Yao, Yehao Li, Yu Wang, Chong-Wah Ngo, and Tao Mei. Transferrable prototypical networks for unsupervised domain adaptation. In *CVPR*, pages 2239–2247, 2019.

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

[30] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *CVPR*, pages 4161–4170, 2017.

[31] Jeany Son. Contrastive learning for space-time correspondence via self-cycle consistency. In *CVPR*, pages 14679–14688, 2022.

[32] Jie Song, Limin Wang, Luc Van Gool, and Otmar Hilliges. Thin-slicing network: A deep structured model for pose estimation in videos. In *CVPR*, pages 4220–4229, 2017.

[33] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, pages 8934–8943, 2018.

[34] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pages 402–419, 2020.

[35] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, pages 7167–7176, 2017.

[36] Jack Valmadre, Luca Bertinetto, Joao Henriques, Andrea Vedaldi, and Philip HS Torr. End-to-end representation learning for correlation filter based tracking. In *CVPR*, pages 2805–2813, 2017.

[37] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *CVPR*, pages 9481–9490, 2019.

[38] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *ECCV*, pages 391–408, 2018.

[39] Ning Wang, Yibing Song, Chao Ma, Wengang Zhou, Wei Liu, and Houqiang Li. Unsupervised deep tracking. In *CVPR*, pages 1308–1317, 2019.

[40] Ning Wang, Wengang Zhou, and Houqiang Li. Contrastive transformation for self-supervised correspondence learning. In *AAAI*, pages 10174–10182, 2020.

[41] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, pages 2566–2576, 2019.

[42] Ni Xiao and Lei Zhang. Dynamic weighted learning for unsupervised domain adaptation. In *CVPR*, pages 15242–15251, 2021.

[43] Enze Xie, Jian Ding, Wenhai Wang, Xiaohang Zhan, Hang Xu, Peize Sun, Zhenguo Li, and Ping Luo. Detco: Unsupervised contrastive learning for object detection. In *ICCV*, pages 8392–8401, 2021.

[44] Jia Xu, René Ranftl, and Vladlen Koltun. Accurate optical flow via direct cost volume processing. In *CVPR*, pages 1289–1297, 2017.

[45] Jiarui Xu and Xiaolong Wang. Rethinking self-supervised correspondence learning: A video frame-level similarity perspective. In *ICCV*, pages 10075–10085, 2021.

[46] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018.

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

[48] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. Fully convolutional adaptation networks for semantic segmentation. In *CVPR*, pages 6810–6818, 2018.

[49] Zixu Zhao, Yueming Jin, and Pheng-Ann Heng. Modelling neighbor relation in joint space-time graph for video correspondence learning. In *ICCV*, pages 9960–9969, 2021.