Contrastive Learning for Space-time Correspondence via Self-cycle Consistency

Jeany Son
AI Graduate School, GIST
Gwangju, South Korea
jeany@gist.ac.kr

Abstract

We propose a novel probabilistic method employing Bayesian Model Averaging and self-cycle regularization for spatio-temporal correspondence learning in videos within a self-supervised learning framework. Most existing methods for self-supervised correspondence learning suffer from noisy labels that come with the data for free, and the presence of occlusion exacerbates the problem. We tackle this issue within a probabilistic framework that handles model uncertainty inherent in the path selection problem built on a complete graph. We propose a self-cycle regularization to consider a cycle-consistency property on individual edges in order to prevent converging on noisy matching or trivial solutions. We also utilize a mixture of sequential Bayesian filters to estimate posterior distribution for targets. In addition, we present a domain contrastive loss to learn discriminative representation among videos. Our algorithm is evaluated on various datasets for video label propagation tasks including DAVIS2017, VIP and JHMDB, and shows outstanding performances compared to the state-of-the-art self-supervised learning based video correspondence algorithms. Moreover, our method converges significantly faster than previous methods.

1. Introduction

With recent advances in deep neural networks and contrastive learning, there have been rapid and substantial progress in self-supervised visual representation learning [3, 10, 11, 13, 15, 16, 33, 35]. This approach explores implicit supervisory patterns capture in massive unlabeled images or videos. Recently, it has succeeded to learn much richer representations compared to the strong supervised learning approaches. Yet, it has not lead to advances in learning temporal correspondences from video.

Learning representations for visual correspondence across space and time is a one of fundamental problems in computer vision, and closely related to many vision tasks, such as video object tracking [1, 40, 46], video object segmentation [2, 32, 43], and optical flow estimation [9, 21]. The collection of large-scale visual correspondence dataset, however, is dreadfully labor-intensive, and the quality of labels depends on human annotators. Recently, self-supervised visual correspondence learning becomes one solution to handle this problem by leveraging a large amount of raw videos for training [22, 23, 29–31, 44, 47, 49, 53]. This self-supervised visual correspondence problem is formulated by a query-target matching problem in videos, which finds an affinity between the query frame and the target frame to match corresponding points.

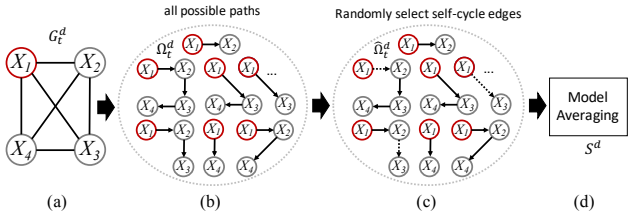


Figure 1. Procedure of proposed model-averaged filtering with multiple paths. Our model conducts a (a) complete graph from given a video clip, and (b) finds palindrome sequences starting from the first frame (red circles). (c) Self-cycle edges (dashed arrows) are selected in a stochastic manner. (d) All chain models are averaged to obtain the final posterior.

Most of recent advanced methods, however, rely on noisy labels captured from colors [29, 30, 44], cycle-consistency constraint [22, 31, 47, 49, 53], positive/negative sample mining [23], or frame-level similarity [50] during training. Directions of these approaches are promising, but they often rely on sophisticated manners or ambiguous supervisions that may lead to falling in local optima and overfitting to the noisy labels. To alleviate this problem, a probabilistic method [22] is proposed to find paths through the graph by performing a random walk between query and target nodes. The problem, however, is that it may suffer from occlusions since the graph in [22] is modeled by the first-order Markov chain where the edge is linked between two consecutive frames. The very recent approach [23] tackles

those issues by collecting well-defined samples by solving optimal transport problem and measuring their uncertainties. Although this method tries to handle uncertainties by collecting positive samples, it still suffers from ambiguity of labels since positive and negative samples are selected by their similarity scores only.

Another limitation of most self-supervised correspondence learning methods [22, 23, 29, 30, 49, 53, 53] in videos is that they focus only on contrastive learning within a given video. Intra-inter consistency loss in [47] is proposed to encourage both positive feature invariance and negative video embedding separation, but it uses an indirect reconstruction loss rather than a contrastive loss, and its performance is far from the state-of-the-art one. Meanwhile, [50] focuses on contrastive learning at video frame-level representations. This method, however, only discriminates the coarse frame-level representations within and across the videos without considering fine patch-level representations.

We propose a new self-cycle regularization method that implies a cycle-consistency constraint for each edge in the graph, which help preventing an overfitting by incorrect matching at the early stage on training. This simple self-cycle edge preserves a cycle-consistency constraint within individual edges as well as a whole path. However, since it is intractable to handle all possible paths having self-cycle edges, we use stochastic sampling to reduce complexity and address lack of model diversity.

In addition, we propose a novel Bayesian framework that learns visual representations for dense correspondences using a complete space-time graph from an input video. The complete space-time graph is constructed from a video clip where nodes correspond to grid patches in frames, and edges connect two nodes between all different frames. We then extract a path set including all possible paths starting from the first frame and returning back to the first frame again. This is called the palindrome sequence as in [22]. Since we cannot identify where occlusions exist during training, we leverage the Bayesian Model Averaging (BMA) [18] to handle uncertainty inherent in the model selection process. By averaging all possible competing paths by BMA, we can alleviate overfitting and uncertainty problems in the model. In Figure 1, we illustrate a simplified procedure of our model-average filtering on multiple paths extracted on a complete graph.

Furthermore, we present a domain contrastive loss that discriminates learned representations of different videos in the embedding space. Most methods typically focus on learning similar representation for positive matches and distant one for negative matches within a video. In order to learn more comprehensive representations for multiple domains which correspond to videos, we aggregate all features within a video and then apply the domain-wise contrastive loss to distinguish them. This can be viewed as coarse-to-fine contrastive learning for videos.

Our contributions are summarized as follows:

- We propose self-cycle edges that implicitly deal with cycle-consistency in a single edge and mitigate a problem of falling into local optima in earlier training stage.
- A mixture of sequential Bayesian filters is used to formulate space-time correspondences on multiple paths considering multi-hops in a complete graph constructed from a video. It can handle model uncertainty by considering all paths in the graph simultaneously.
- Our batch-wise domain contrastive loss discriminates negative pairs not only in the same video but also between different videos. It leads to learn fine-grained visual representations for dense correspondences.
- Our method not only outperforms the state-of-the-art algorithms but also converges much faster on various video benchmarks.

2. Related Works

Self-supervised Visual Representation Learning Self-supervised learning supports to learn task-agnostic visual representations with different pretext tasks. Early works for self-supervised representation learning have focused on pretext tasks exploiting some property of the data [3, 8, 10, 28, 35, 48] without any human supervision. Recent works focs on contrastive learning using Siamese network [4, 5, 13, 16] that encourages positive image pairs to be close together and negative image pairs to stay away from each other. Recently, the representations captured by self-supervised learning have outperformed supervised ones on several downstream tasks. The method that firstly has outperformed the supervised one in multiple downstream recognition tasks is MoCo [16], where a momentum network is used to encode the large number of negatives as well as a positive one by contrastive learning. BYOL [13] have utilized a diverse data augmentation technique to enlarge the size of datasets. In this paper, we adopt this image-level contrastive learning in our new domain contrastive loss.

Space-Time Dense Correspondence Learning in Videos Most recently, several works have focused on self-supervised learning to find dense correspondences in unlabeled videos [22, 23, 29–31, 44, 50, 53]. The first attempt was to learn temporal correspondence between two frames by comparing original colors of the future frame with propagated colors from the current frame to the future frame by their affinity [44]. Followed by this, CorrFlow [30] proposed a restricted attention [30] which reduces computational costs of computing affinity, and MAST [29] proposed memory-augmented learning [29] utilizing memory bank.

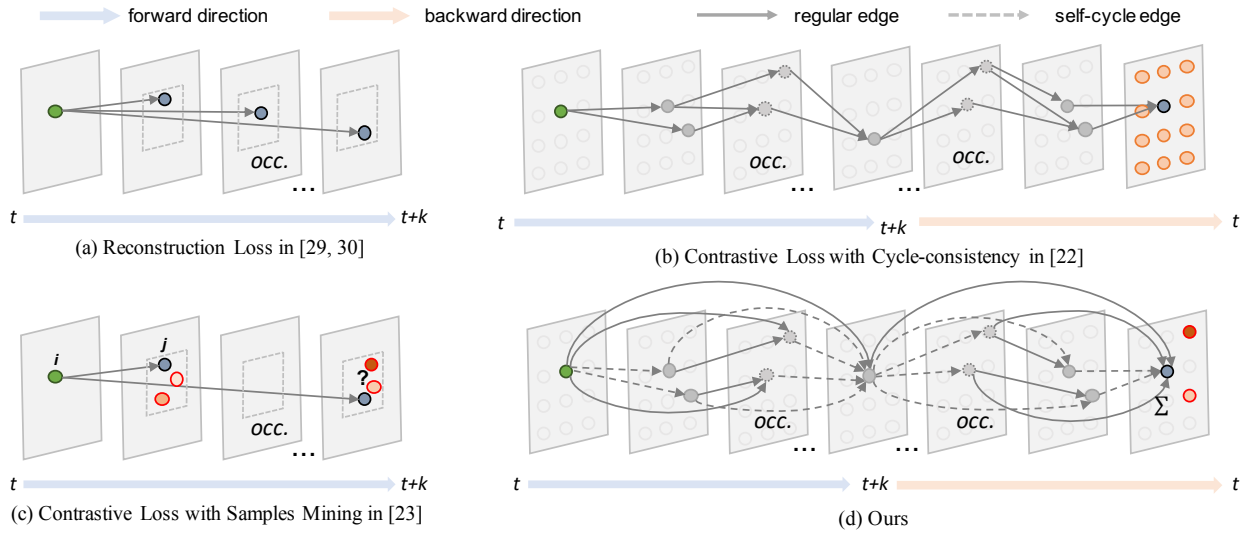


Figure 2. Comparison with self-supervised correspondence learning methods in terms of their losses and correspondences between frames. Green, blue, and red circles denote queries, positive samples, and negative samples, respectively. Darker reds denote more hard negative samples. Dashed boxes in (a) and (c) denotes local search windows for training. Our method takes advantages of both cycle-consistency and hard negatives in consideration of multi-hop concurrent paths.

Those works have used colors as their self-supervisions and the goal of the task is to reconstruct colors. Concurrently, methods focusing on reconstructing images using auto-encoder have proposed [31, 47]. Another approach to learn visual representations has used cycle-consistency constraint [22, 49, 53] as self-supervision where good correspondences should match points bi-directionally. [22] proposed a probabilistic framework utilizing contrastive random walk where pairwise similarity defines transition probability of a random walk between two consecutive frames. Recently, [23] proposed a method collecting good positive samples by solving optimal transport problem, and hard negative ones by controlling their hardness during training. In contrast, our method deals with all pairs of frames to compute optimal posterior by model averaging, and handles hard negatives within a cycle-consistency constraint. Moreover, We focus on discriminating representations across videos in order to learn course-to-fine similarity. We illustrate the major difference between recent advanced methods compared and our method in Figure 2.

Regularization on deep learning Deep neural network often suffers from an overfitting problem arising from its nature of overparametrization. There have been numerous methods that focus on alleviating this problem by regularization techniques [7, 20, 27, 34, 38, 39, 42, 51, 52]. In particular, dropout [42], dropconnect [45], and dropedge [39] have employed binary random selection to hidden units or connections of neural networks or graphs. Learning with stochastic depth [14, 20] can be interpreted as a regularization method by noise injection into model architecture. Data augmentation [7, 51, 52] has been also used to avoid overfitting. In this paper, we focus on regularization to alleviate an overfitting problem due to noisy matching and early convergence to a trivial solutions.

3. Method

This section describes our probabilistic framework based on Bayesian Model Averaging (BMA) and self-cycle regularization for space-time correspondence learning. In Figure 3, the overall framework of our method is described.

3.1. Algorithm Overview

We first start with constructing a complete graph in space and time domain to handle multiple hypothesis paths and existence of occlusions. A complete graph $G_t^d = (V_t^d, E_t^d)$ is constructed with the K number of key frames starting from t in a given video domain d, where each node in the graph is connected to all other nodes in all key frames in videos, $V_t^d = \cup_{k=1}^{K} X_{t+k-1}^d$, and X_k^d denotes the k-th frame in a video. Once a complete graph, G_t^d, is constructed, a set of all possible palindrome paths Ω_t^d on G_t^d is extracted by

$$\Omega_t^d = \{p | p = ([v_1, \dots, v_{k_{l-1}}], v_{k_l}, \text{rev}([v_1, \dots, v_{k_{l-1}}])), \\ v_{k_l} \in X_{t+k_l-1}^d,\ k_l > k_{l-1},\ 1 < l \leq k\}, \quad (1)$$

where v_{k_l} is the node in the X_{t+k_l-1} frame, k_l is a keyframe index, and $\text{rev}([\cdot])$ is the function that returns a list in a reverse order of the given list. This set only contains paths

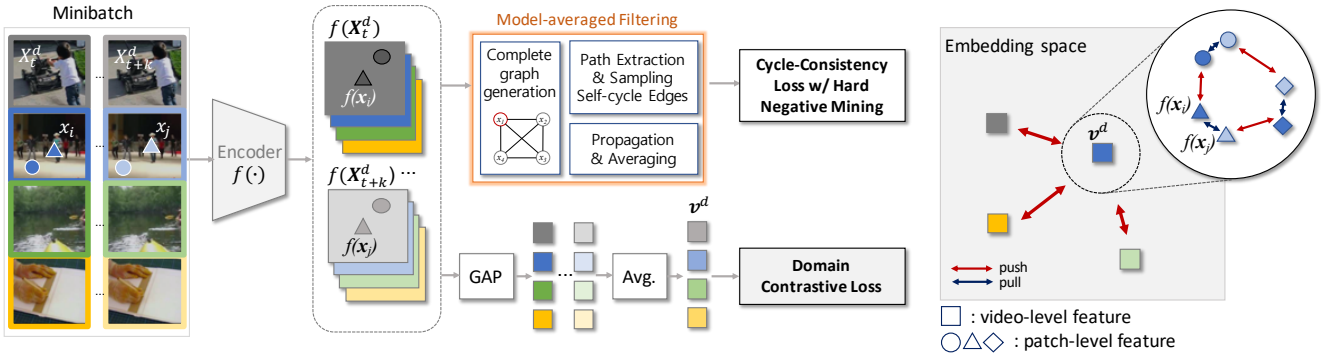


Figure 3. Overall framework for training our network using proposed losses. Our minibatch consists of D number of different videos with K number of keyframes for each video.

in a monotonic and palindromic order starting at the first frame in a given video clip. For examples, a path $p \in \Omega_t^d$ can be a sequence of $(v_1, v_2, v_4, v_2, v_1)$, but cannot be either $(v_1, v_4, v_3, v_4, v_1)$ or (v_2, v_3, v_2).

After extracting all possible paths Ω_t^d in the graph G_t^d, we randomly select self-cycle edges in the given paths where the probability of selection is drawn from Bernoulli distribution. This self-cycle edge has a cycle between two nodes linked by a given edge. (We will discuss this in Section 3.3)

We then follow [22] to construct an affinity matrix. Each element of an affinity matrix, $\mathcal{A}^{(t,t')} \in \mathbb{R}^{n\times n}$, between two frames X_t and $X_{t'}$, which is a pairwise similarity by applying a softmax function with temperature τ, is given by

$$\begin{aligned}\mathcal{A}_{ij}^{(t,t')} &= P(X_{t'}^d = j | X_t^d = i) \\ &= \frac{\exp(f(\mathbf{x}_i)^T f(\mathbf{x}_j)/\tau)}{\sum_j \exp(f(\mathbf{x}_i)^T f(\mathbf{x}_j)/\tau)},\end{aligned} \tag{2}$$

where $\mathbf{x}_i \in X_t^d$ and $\mathbf{x}_j \in X_{t'}^d$ are nodes in frame X_t and frame $X_{t'}$, respectively, n is the number of nodes in the frame, i, j are the node indexes, and $f(\cdot)$ denotes an encoder network. Note that, our affinity is computed not only between two consecutive frames but also frames that are far apart from each other. Then, by utilizing Bayesian model averaging, we compute posterior probability in the last frame which is identical to the first frame by considering all paths in Ω_t^d . (This will be covered in Section 3.2.)

The whole encoder network is then trained using a cycle-consistency contrastive loss with hard negatives and a domain contrastive loss as shown in Figure 3. These loss functions will be discussed in Section 3.4

3.2. Bayesian Model Averaging

We design all palindrome paths p in Ω_t^d as a Markov chain model. However, selecting the best one out of all possible paths is not straightforward since it is uncertain that which one is good for matching. For this reason, we adopt a model averaging strategy [18, 19], where the final posterior is determined by averaging estimate of all possible chain models instead of choosing one.

Let $\mathbf{y}_k$ be the target state in the k-th frame in Ω_t^d, where k is the last node as well as the first frame in the video. Given all possible paths $p \in \Omega_t^d$, by Bayesian model averaging strategy, the posterior of $\mathbf{y}_k$ is given by

$$\bar{P}(\mathbf{y}_k) = \sum_{p\in\Omega_t^d} P(\mathbf{y}_k | p, \mathbf{z}_k) P(p), \tag{3}$$

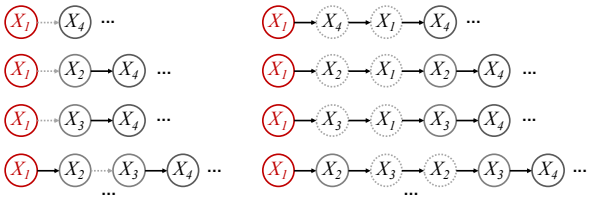
where $\mathbf{z}_k = (z_1, \ldots, z_k)$ is an observation variable, and we assume that prior of the paths is given by a uniform distribution, $P(p) = 1/|\Omega_t^d|$. Since each palindrome path is modeled by the first order Markov chain, we can use Bayes theorem as follow:

$$\begin{aligned}&P(\mathbf{y}_k | p, \mathbf{z}_k) \\ &\propto \int P(z_k|\mathbf{y}_k) P(\mathbf{y}_k|\mathbf{y}_l) P(\mathbf{y}_l | p_l, \mathbf{z}_l) d\mathbf{y}_l.\end{aligned} \tag{4}$$

where, l denotes the second last element in p. We can now estimate the posterior of $\mathbf{y}_{t_k}$ in a simple and recursive fashion, and we approximate it by sampling as which is formally given by

$$\begin{aligned}\bar{P}(\mathbf{y}_k) &\propto \frac{1}{|\Omega_t^d|} \sum_{p\in\Omega_t^d} P(\mathbf{z}_k|\mathbf{y}_k) \int P(\mathbf{y}_k|\mathbf{y}_l) \bar{P}(\mathbf{y}_l) d\mathbf{y}_l \\ &\approx \frac{1}{|\Omega_t^d|} \sum_{p\in\Omega_t^d} \sum_{\mathbf{y}_l^i \in \mathbb{S}_l} P(\mathbf{z}_k|\mathbf{y}_k) P(\mathbf{y}_k|\mathbf{y}_l^i),\end{aligned} \tag{5}$$

where $\mathbb{S}_l$ denotes a set of grid samples drawn from $\bar{P}(\mathbf{y}_l)$. Since our method does not have any predefined appearance model, the transition model and likelihood are jointly de-



(a) Possible paths in $X_1 \rightarrow X_4$ (b) unrolled self-cycle edge

Figure 4. Examples of self-cycle edges by unrolling the edge to have a sequential order. Dashed arrows indicate self-cycle edges. The paths in (a) are equivalent to ones in (b). Those self-cycle edges are selected from Bernoulli distribution.

fined by affinity between two frames given p as follows,

$$\sum_{\mathbf{y}_l^i \in \mathbb{S}_l} P(z_k|\mathbf{y}_k)P(\mathbf{y}_k|\mathbf{y}_l^i) = (\prod_{(u,v)\in\mathcal{E}_p\setminus(l,k)} \mathcal{A}^{(u,v)})\mathcal{A}^{(l,k)}, \quad (6)$$

where $\mathcal{E}_p = \{(u,w)|u = v_l, w = v_{l+1}, l = (1,\dots,|p|-1)\}$ is an edge set for a path p. We then rewrite Eq. (5) using Eq. (6) to obtain a matrix form of the final posterior probability regarding all grid locations for a given Ω_t^d which corresponds to the similarity matrix $S^d \in \mathbb{R}^{n\times n}$:

$$S^d = \bar{P}(\mathbf{y}_k) \approx \frac{1}{|\Omega_t^d|} \sum_{p\in\Omega_t^d} \prod_{(u,v)\in\mathcal{E}_p} \mathcal{A}^{(u,v)}, \quad (7)$$

where n is the number of nodes in each frame and d denotes a video index.

Although our method relies only on an affinity matrix between frames, the problem due to occlusions or drifting can be alleviated by aggregating all the propagated densities through Bayesian model averaging. This leads to reduce the risk of overfitting in the presence of extreme noise in self-supervisory during training. More detailed derivations are provided in our supplementary material.

3.3. Self-cycle Regularization

As we mentioned in Section 3.1, we define a self-cycle edge that additionally includes backward and forward paths for a given edge:

$$\hat{\mathcal{A}}^{(u,v)} = \begin{cases} \mathcal{A}^{(u,v)}\mathcal{A}^{(v,u)}\mathcal{A}^{(u,v)}. & \text{if } \alpha = 1 \\ \mathcal{A}^{(v,u)} & \text{otherwise,} \end{cases} \quad (8)$$

where α is a binary random variable. However, constructing a complete graph including all self-cycle edges is computationally expensive. For example, if we have $|E|$ number of edges in a given path $p \in \Omega_t^d$, then the total number of paths taking into account cycles is $2^{|E|}$. Thus, due to the computational limitation, we randomly select self-cycle edges among all edges in the path p using α. Specifically, if we assume that there exist e_p edges in the path p, the binary random variable $\alpha_e (e = 1,\dots,e_p)$ is obtained by

$$\alpha_e \sim \text{Bernoulli}(\beta_e), \quad (9)$$

where β_e is the parameter of Bernoulli distribution corresponding to the e-th edge. The binary variable α_e indicates whether the e-th edge is to be selected for an edge having a self-cycle. Figure 4 shows examples of paths with self-cycle edges and their unrolling paths having sequential order.

We then rewrite Eq. (7) considering self-cycle edges as follows

$$S^d = \bar{P}(\mathbf{y}_k) \approx \frac{1}{|\hat{\Omega}_t^d|} \sum_{p\in\hat{\Omega}_t^d} \prod_{(u,v)\in\mathcal{E}_p} \hat{\mathcal{A}}^{(u,v)}, \quad (10)$$

where $\hat{\Omega}_t^d$ is a set of all possible paths in which self-cycle edges are selected by stochastic sampling. This simple cycle preserves a cycle-consistency constraint within intermediate edges as well as a whole path, and mitigates overfitting problems.

3.4. Losses for Self-supervised Learning

Our goal is now to learn dense correspondences in the large amount of raw videos without human supervisions. Our network is trained using the loss function:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cyc\_hn}} + \mathcal{L}_{\text{domain}}, \quad (11)$$

where $\mathcal{L}_{\text{cyc\_hn}}$ and $\mathcal{L}_{\text{domain}}$ denote a cycle-consistency contrastive loss with hard negatives and a domain contrastive loss, respectively.

Cycle-Consistency Loss with Hard Negatives For training, we follow [22] that explores cycle-consistency in time, where the target of the query node should be located in its original position. In addition to a cycle-consistency constraint, we use hard negative mining strategy to avoid trivial solutions during training. Since the target of positive sample should be its original position, diagonal elements in S^d become positive samples. For mining hard negatives, akin to [23], we define semi-hard negatives using their ranks as

$$\mathcal{N}_i = \{S_{ij}^d | m_1 < r(S_{ij}^d) < m_2, i \neq j\}, \quad (12)$$

where $r(\cdot)$ computes a normalized rank of S_i^d which is the i-th row in S sorted in descending order. Then our cycle-consistency loss with hard negatives is defined as

$$\mathcal{L}_{\text{cyc\_hn}} = \sum_d \sum_i -\log \frac{\exp(S_{ii}^d)}{\exp(S_{ii}^d) + \sum_{|\mathcal{N}_i|} \exp(S_{i\mathcal{N}_i}^d)}, \quad (13)$$

where d is the index of a video. Note that, unlike [23], we do not use any sophisticated curriculum learning for negative mining.

Table 1. Quantitative results on the DAVIS2017 [37] validation set. We illustrate the results of the state-of-the-art self-supervised correspondence learning methods, image-level self-supervised contrastive learning method, as well as some supervised methods in comparison of our method. We also show backbones and datasets used for training. † denotes that backbone networks are initially pre-trained on ImageNet [6] dataset. Attributes are provided for several types for self-supervisions, where C = Color (RGB or Lab), F = Features, I = Images, P = Position, S = Strong supervision.

Method	Backbone	Dataset	Attribute	$\mathcal{J}\&\mathcal{F}_{mean}$	$\mathcal{J}_{mean}$	$\mathcal{J}_{recall}$	$\mathcal{F}_{mean}$	$\mathcal{F}_{recall}$
Colorization [44]	ResNet-18	Kinetics	C	34.0	34.6	34.1	32.7	26.8
CorrFlow [30]	ResNet-18	OxUvA	C	50.3	48.4	53.2	52.2	56.0
MAST [29]	ResNet-18	YT-VOS	C	65.5	63.3	73.2	67.6	77.7
UVC [31]	ResNet-18	Kinetics	I/P	60.9	59.3	68.8	62.7	70.9
CRW [22]	ResNet-18	Kinetics	F	67.6	64.8	76.1	70.2	82.1
ContrastCorr [8]	ResNet-18	Kinetics	I/F	63.0	60.5	-	65.5	-
VFS-18 [50]	ResNet-18	Kinetics	F	66.7	64.0	-	69.4	-
NRG [53]	ResNet-18	Kinetics	F/P	68.7	65.8	77.7	71.6	84.3
MBS [23]	ResNet-18	YT-VOS	F	70.3	**67.9**	78.2	72.6	83.7
Ours	ResNet-18	Kinetics	F	**70.5**	67.4	**78.8**	**73.6**	**84.6**
TimeCycle [49]	ResNet-50	VLOG	F/P	48.7	46.4	50.0	50.0	48.0
VFS-50 [50]	ResNet-50	Kinetics	F	68.9	66.5	-	71.3	-
OSVOS [2]	VGG-16	DAVIS†	S	60.3	56.6	63.8	63.9	73.8
FEELVOS [43]	Xception-65	COCO/DAVIS/YT-VOS†	S	71.5	69.1	79.1	74.0	83.8
STM [36]	ResNet-50	DAVIS/YT-VOS†	S	81.8	79.2	-	84.3	-
VINCE [12]	ResNet-50	Kinetics	F	60.4	57.9	66.2	62.8	71.5
MoCo [16]	ResNet-50	ImageNet	F	65.4	63.2	73.0	67.6	78.7

Domain Contrastive Loss Since our cycle consistency loss only considers representation learning within a single video, we further propose a new loss for separating learned representations between videos. In order to obtain holistic representations for a video d, we apply a Global Average Pooling (GAP) to final output features of given frames, and average these features with respect to frames:

$$\mathbf{v}^d = \frac{1}{K}\sum_{k=0}^{K-1}(\text{GAP}(f(\mathbf{X}^d_{t+k}))), \tag{14}$$

where $\text{GAP}(\cdot)$ is a global average pooling operation, We then compute similarity $\hat{S}$ between all videos as

$$\hat{S}_{dd'} = \frac{\exp(\langle \mathbf{v}^d, \mathbf{v}^{d'} \rangle / \tau)}{\sum_{d'} \exp(\langle \mathbf{v}^d, \mathbf{v}^{d'} \rangle / \tau)},\ \hat{S} \in \mathbb{R}^{D \times D}, \tag{15}$$

where τ is a temperature hyper-parameter as same in Eq. (2). Finally, our domain contrastive loss $\mathcal{L}_{\text{domain}}$ is defined as follows:

$$\mathcal{L}_{\text{domain}} = \sum_{d} -\log \frac{\exp(\hat{S}_{dd})}{\sum_{d'} \exp(\hat{S}_{dd'})}. \tag{16}$$

4. Implementation Details

Encoder We adopt ResNet-18 [17] network architecture as our backbone where the strides of last two residual blocks are modified to one in order to increase the spatial resolution of the feature map and fuse the last two layer. Then we apply linear projection and l_2 normalization to compute the final embedding vector.

Training The input image size is 256×256 for training, where the size of patches are 64×64 sampled on 7×7 grid, like [22]. The input images and patches are spatially jittered. We apply Edge Dropout to the transition matrix $\mathcal{A}$ and re-normalize followed by [22] with rate 0.1. We use the Kinetics dataset [25] for self-supervised training which consists of 240k training videos. We train our network using the Adam optimizer [26] with 25 epochs and a learning rate of 10^{-4}. Temperature τ is set to 0.05 in Eq. (2). We threshold affinity by 0.5. The number of keyframes K for training is set to 4 with a framerate of 8. A self-cycle edge is selected based on α_e from a Bernoulli distribution with $\beta_e = 0.5$. The ranges for negative mining are set to $m_1 = 0.6$, $m_2 = 0.9$ in Eq. (12). Note that, in our experiments, most of hyper-parameters are followed by CRW [22] which is the baseline of our method.

Inference We follow the testing protocols in [22] for all tasks. In order to evaluate the learned representation, we test on several video label propagation tasks, where ground-truths of the targets in a video are given at the first frame. Those labels are propagated to the rest of the frames by affinity, where propagated labels at the next frame are computed by a weighted average of labels at the previous frame with their affinity scores.

5. Experiments

We evaluate our learned representation on various dense correspondence benchmarks including DAVIS2017 [37], JHMDB [24], and VIP [54], which are tasks for video object segmentation, human pose key-point tracking and hu-

Table 2. Quantitative results on the DAVIS2017 test-dev set.

	Sup.	$\mathcal{J}\&\mathcal{F}_{mean}$	$\mathcal{J}_{mean}$	$\mathcal{F}_{mean}$
CRW [22]	✗	55.9	52.3	59.6
VFS [50]	✗	57.3	53.1	61.6
Ours	✗	**59.9**	**55.9**	**64.0**
OSVOS [2]	✓	50.9	47.0	54.8
FEELVOS [43]	✓	57.8	55.1	60.4

Table 3. Quantitative results for Human Part segmentation and Pose Tracking tasks on VIP and JHMDB datasets, respectively.

Method	Sup.	VIP [54]	JHMDB [24]	
		mIoU	PCK@0.1	PCK@0.2
UVC [31]	✗	34.1	58.6	79.6
CRW [22]	✗	38.6	59.3	84.9
ContrastCorr [8]	✗	37.4	61.1	80.8
MBS [23]	✗	37.8	60.5	82.3
VFS-18 [50]	✗	39.9	60.5	79.5
NRG [53]	✗	40.2	61.4	**85.3**
Ours	✗	**40.8**	**61.7**	82.6
ATEN [55]	✓	37.9	-	-
TSN [41]	✓	-	68.7	92.1

man body-part propagation, respectively. Our method is compared with the state-of-the-art dense correspondence algorithms based on self-supervised learning, including Colorization [44], CorrFlow [30], MAST [29], TimeCycle [49], UVC [31], CRW [22], VFS [50], NRG [53], and MBS [23]. We also compare our model with several supervised video object segmentation algorithms, including OSVOS [2], FEELVOS [43] and STM [36]. In addition, image- and video-level self-supervised contrastive learning methods including MoCo [16] and VINCE [12] are compared. ATEN [55] and TSN [41] are compared for human part segmentation and pose tracking tasks on VIP and JHMDB datasets, respectively.

5.1. Results

Video Object Segmentation We first compare our dense temporal correspondence results to the state-of-the-art methods on the validation set of DAVIS2017 [37], which is the most popular benchmark for a video object segmentation task. The segmentation masks for the multiple target objects are given at the first frame, then the outputs for this task is to find the segmentation masks for the targets in the rest of frames in video. The performances are evaluate by the mean and recall of Jaccard Index $\mathcal{J}$ and contour accuracy $\mathcal{F}$ following the benchmark protocol. $\mathcal{J}\&\mathcal{F}_{\text{mean}}$ is an average of $\mathcal{J}$ and $\mathcal{F}$. To evaluate on DAVIS2017 validation set, we use the resolution of 480p for an input image, We report $\mathcal{J}$, $\mathcal{F}$ performances as well as several attributes for the state-of-the-art self-supervised methods in Table 1.

As shown in Table 1, our method outperforms all other state-of-the-art self-supervised methods. We also compare our method to regular self-supervised contrastive learning methods such as MoCo [16] and VINCE [12] as well as

Table 4. Ablation studies for our model on the DAVIS2017 val dataset. DCL, SC, BMA and HN denote domain contrastive loss, self-cycle edges, Bayesian model averaging and hard negatives, respectively.

a) DCL	b) SC	c) BMA	d) HN	$\mathcal{J}\&\mathcal{F}_{mean}$
✓	✓	✓	✓	70.5
✓	✓	✓	✗	70.1
✓	✓	✗	✗	69.4
✓	✗	✗	✗	68.0
✗	✗	✗	✗	67.6

several strong supervised approaches for the video object segmentation task. Our method surpass some supervised methods such as OSVOS [2]. Also, the performance of our method is similar to FEELVOS [43] which is trained with a large amount of densely annotated datasets. In Figure 5, we illustrate our video object segmentation results in the fist two rows.

We also evaluate our method on DAVIS2017 test-dev set, and report the results in Table 2. It clearly shows that our method also outperforms CRW, VFS and some fully supervised methods [2, 43] with a large margin.

Video Part Segmentation We also evaluate our method on a video part segmentation task using Video Instance Parsing (VIP) [54] dataset, which contains 20 parts of human body. The image size is resized to resolution of 560p. The evaluation metrics are mean intersection-over-union (IoU) which is equivalent to $\mathcal{J}_{\text{mean}}$. As shown in Table 3, our method outperforms other stat-of-the-art self-supervised learning based methods as well as a strong supervised method [55].

Human Pose Tracking To evaluate our methods on JHMDB [24] dataset for human keypoints tracking. We follow the protocol of [22], resizing the input image into 320p resolution images for testing. We use the evaluation metric of the possibility of correct keypoints (PCK) [41] which computes the percentage of keypoints close to ground-truth in different thresholds (e.g. 0.1 or 0.2). In Table 3, we show that the performance of our model surpasses others in PCK@0.1 metric.

5.2. Ablation Study on DAVIS2017

Convergence Analysis We show the performances over 0.5M training iterations in comparison with CRW [22] and MBN [23] in Figure 6a. Our method converges significantly faster than other two methods. We also show the convergence rate on variants of our method in Figure 6b. In particular, our method achieves 67.6% within two epoch which is the best performance of [22]. We borrow the graph in [23] to compare convergence rate.

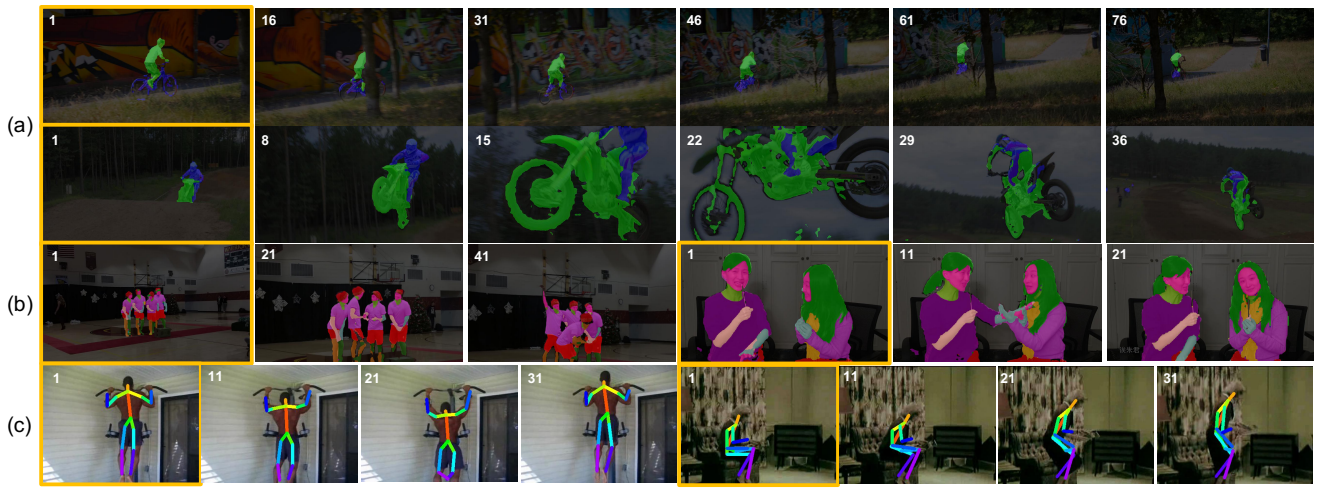
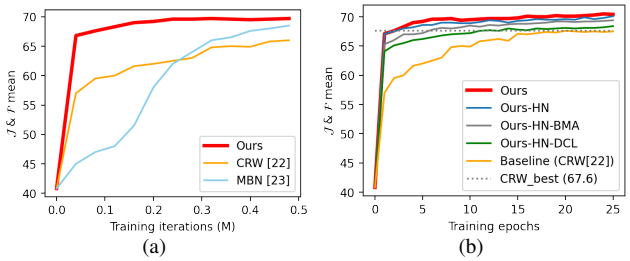
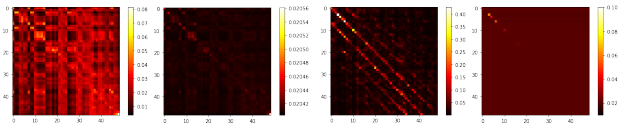


Figure 5. Qualitative results of our method on several label propagation tasks. (a) Video object segmentation on DAVIS2017 [37] dataset, (b) Human part propagation on VIP [54] dataset, (c) Human pose tracking on JHMDB [24] dataset.



(a) (b)

Figure 6. Convergence Analysis on DAVIS2017 [37] validation set. We show performance convergence plots compared to (a) other SOTA methods [22, 23] and (b) variants of our method.



(a) (b) (c) (d)

Figure 7. Visualization of affinity between two frames. (a) and (b) are affinities before training without and with a self-cycle edge, respectively. (c) and (d) shows affinities after training without and with a self-cycle edge, respectively.

Component Analysis We analyze each component of our model in Table 4. Our self-cycle edges improve the performance by 1.4% in $\mathcal{J}\&\mathcal{F}_{\text{mean}}$, and our domain contrastive loss by 0.4%. By employing Bayesian model averaging on multiple paths and hard-negative mining, the performance is improved by 0.7% and 0.4%, respectively. This clearly shows that each component of our method contributes to performance improvements.

Effect of Self-cycle Edges In Figure 7, we visualize affinities to show the effect of our self-cycle edges. As we shown in (a) and (c) in Figure 7, affinities without self-cycle edges have many noises even after training. However, when we use self-cycle edges, the affinities become more sparse but smoothed. This effect is similar to solving optimal transport in [23]. While [23] solves optimal transport problem to find the best matching point for a positive sample, our self-cycle edges regularize the affinity to consider multiple paths as many as possible. We observe that this strategy is helpful to improve accuracy and training convergence. We believe that the slow convergence rate of CRW [22] and MBS [23] than ours may come from this ambiguous matching at the early stage of training. With regularized affinity scores, our self-cycle edges can have an ability to avoid overfitting from noisy matching and prevent a trivial solutions. This simple idea also allows affinity to take into account a cyclic consistency in a single edge.

6. Conclusions

We present a new method for self-supervised dense correspondence learning in video within a probabilistic framework. We adopt a mixture of sequential Bayesian filters on multiple paths with self-cycle edges on a video to handle uncertainty problems due to the nature of self-supervised learning. Proposed domain contrastive loss also contribute to performance gains by regularizing and fine-grained learning. The outstanding performances and an extremely faster convergence rate in training are proved in intensive experiments on various label propagation tasks.

Acknowledgement. This work was supported by the IITP grants (No.2019-0-01842, No.2021-0-02068) funded by the Korea government (MSIT), the GIST-MIT Research Collaboration grant funded by the GIST, and the ETRI grant funded by the Korean government [21YR1610].

References

[1] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *ECCV*, 2016. 1

[2] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, 2017. 1, 6, 7

[3] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018. 1, 2

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2

[5] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. 2

[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*, 2009. 6

[7] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 3

[8] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015. 2, 6, 7

[9] 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*, 2015. 1

[10] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE TPAMI*, 38(9):1734–1747, 2015. 1, 2

[11] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 1

[12] Daniel Gordon, Kiana Ehsani, Dieter Fox, and Ali Farhadi. Watching the world go by: Representation learning from unlabeled videos. *arXiv preprint arXiv:2003.07990*, 2020. 6, 7

[13] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Pires, Zhaohan Guo, Mohammad Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020. 1, 2

[14] Bohyung Han, Jack Sim, and Hartwig Adam. Branchout: Regularization for online ensemble tracking with convolutional neural networks. In *CVPR*, 2017. 3

[15] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *ICCVW*, 2019. 1

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1, 2, 6, 7

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. in computer vision and pattern recognition. In *CVPR*, 2016. 6

[18] Jennifer A Hoeting, David Madigan, Adrian E Raftery, and Chris T Volinsky. Bayesian model averaging: a tutorial (with comments by m. clyde, david draper and ei george, and a rejoinder by the authors. *Statistical science*, 14(4):382–417, 1999. 2, 4

[19] Seunghoon Hong, Suha Kwak, and Bohyung Han. Orderless tracking through model-averaged posterior estimation. In *ICCV*, 2013. 4

[20] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016. 3

[21] 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. 1

[22] Allan Jabri, Andrew Owens, and Alexei A Efros. Space-time correspondence as a contrastive random walk. In *NeurIPS*, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[23] Sangryul Jeon, Dongbo Min, Seungryong Kim, and Kwanghoon Sohn. Mining better samples for contrastive learning of temporal correspondence. In *CVPR*, 2021. 1, 2, 3, 5, 6, 7, 8

[24] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *ICCV*, 2013. 6, 7, 8

[25] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, F. Viola S. Vijayanarasimhan, T. Green, T. Back, P. Natsev, and et al. The kinetics human action video dataset. In *arXiv:1705.06950*, 2017. 6

[26] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICML*, 2015. 6

[27] Diederik P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *NeurIPS*, 2015. 3

[28] Nikos Komodakis and Spyros Gidaris. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 2

[29] Zihang Lai, Erika Lu, and Weidi Xie. Mast: A memory-augmented self-supervised tracker. In *CVPR*, 2020. 1, 2, 6, 7

[30] Zihang Lai and Weidi Xie. Self-supervised learning for video correspondence flow. In *BMVC*, 2019. 1, 2, 6, 7

[31] 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*, 2019. 1, 2, 3, 6, 7

[32] Jonathon Luiten, Idil Esen Zulfikar, and Bastian Leibe. Unovost: Unsupervised offline video object segmentation and tracking. In *Proceedings of the IEEE/CVF Winterons of Computer Vision*, 2020. 1

[33] Zelun Luo, Boya Peng, De-An Huang, Alexandre Alahi, and Li Fei-Fei. Unsupervised learning of long-term motion dynamics for videos. In *CVPR*, 2017. 1

[34] Hyeonwoo Noh, Tackgeun You, Jonghwan Mun, and Bohyung Han. Regularizing deep neural networks by noise: Its interpretation and optimization. In *NeurIPS*, 2017. 3

[35] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 1, 2

[36] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019. 6, 7

[37] J. Pont-Tuset, S. Caelles F. Perazzi, P. Arbel´aez, A. Sorkine-Hornung, and L. Van Gool. The 2017 davis challenge on video object segmentation. In *arXiv:1704.00675*, 2017. 6, 7, 8

[38] Russell Reed, Seho Oh, and RJ Marks. Regularization using jittered training data. In *IJCNN*, 1992. 3

[39] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *ICLR*, 2019. 3

[40] Bing Shuai, Andrew Berneshawi, Xinyu Li, Davide Modolo, and Joseph Tighe. Siammot: Siamese multi-object tracking. In *CVPR*, 2021. 1

[41] Jie Song, Limin Wang, Luc Van Gool, and Otmar Hilliges. Thin-slicing network: A deep structured model for pose estimation in videos. In *CVPR*, 2017. 7

[42] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 3

[43] 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*, 2019. 1, 6, 7

[44] Carl Vondrick, Abhinav Shrivastava, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *ECCV*, 2018. 1, 2, 6, 7

[45] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *ICML*, 2013. 3

[46] Ning Wang, Yibing Song, Chao Ma, Wengang Zhou, Wei Liu, and Houqiang Li. Unsupervised deep tracking. In *CVPR*, 2019. 1

[47] Ning Wang, Wengang Zhou, and Houqiang Li. Contrastive transformation for self-supervised correspondence learning. In *AAAI*, 2021. 1, 2, 3

[48] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015. 2

[49] Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019. 1, 2, 3, 6, 7

[50] Jiarui Xu and Xiaolong Wang. Rethinking self-supervised correspondence learning: A video frame-level similarity perspective. In *ICCV*, 2021. 1, 2, 6, 7

[51] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019. 3

[52] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 3

[53] Zixu Zhao, Yueming Jin, and Pheng-Ann Heng. Modelling neighbor relation in joint space-time graph for video correspondence learning. In *ICCV*, 2021. 1, 2, 3, 6, 7

[54] Q. Zhou, X. Liang, K. Gong, and L. Lin. Adaptive temporal encoding network for video instance-level human parsing. In *In Proceedings of the 26th ACM international conference on Multimedia*, 2018. 6, 7, 8

[55] Qixian Zhou, Xiaodan Liang, Ke Gong, and Liang Lin. Adaptive temporal encoding network for video instance-level human parsing. In *ACM MM*, 2018. 7