Modelling Neighbor Relation in Joint Space-Time Graph for Video Correspondence Learning

Zixu Zhao Yueming Jin Pheng-Ann Heng
The Chinese University of Hong Kong
{zxzhao, ymjin, pheng}@cse.cuhk.edu.hk

Abstract

*This paper presents a self-supervised method for learning reliable visual correspondence from unlabeled videos. We formulate the correspondence as finding paths in a joint space-time graph, where nodes are grid patches sampled from frames, and are linked by two type of edges: (i) neighbor relations that determine the aggregation strength from intra-frame neighbors in space, and (ii) similarity relations that indicate the transition probability of inter-frame paths across time. Leveraging the cycle-consistency in videos, our contrastive learning objective discriminates dynamic objects from both their neighboring views and temporal views. Compared with prior works, our approach actively explores the neighbor relations of central instances to learn a latent association between center-neighbor pairs (*e.g.*, "hand – arm") across time, thus improving the instance discrimination. Without fine-tuning, our learned representation outperforms the state-of-the-art self-supervised methods on a variety of visual tasks including video object propagation, part propagation, and pose keypoint tracking. Our self-supervised method also surpasses some fully supervised algorithms designed for the specific tasks.*

1. Introduction

Learning temporal correspondence — a problem of learning "what went where"— is closely related to many fundamental vision tasks, such as video object tracking [46, 26, 45], video object segmentation [48, 4, 44, 29, 32], and flow estimation [7, 14]. In essence, it corresponds to a query-target matching problem, which relies on an affinity to match a physical point (or patch) in the query frame t to that in the target frame $t+k$. One practical issue is collecting dense annotations from large-scale videos, which costs large human efforts. It motivates numerous self-supervised methods [45, 50, 24, 22, 21, 47, 15] to learn dynamic objects from unlabeled videos by leveraging the cycle-consistency in time as a free supervisory signal.

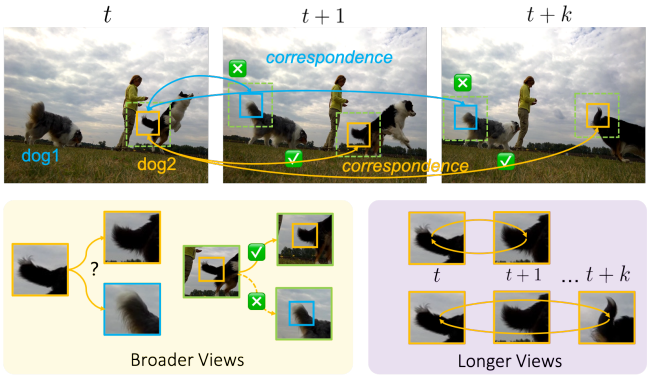

Figure 1. How to find the correspondence of a small object such as the dog tail in a video? We argue that the query-target matching desires both *longer views* (temporal dynamics) and *broader views* (neighbor relations) to distinguish similar instances. We capture these two cues in a graph to learn correspondence.

Recent approaches learn strong representations by constructing long-range views mainly from two perspectives: (i) learning pixel-level correspondences by a *single-step* association [24, 47], or (ii) learning patch-level correspondences by a *multi-step* association [15]. The single-step association can be viewed as a pixel-level affinity between two patches at timesteps t and $t+k$, aiming to transform the pixel colors. Such transformation requires a deterministic correspondence to locate the target patch at $t+k$, which is achieved by training an extra unsupervised patch tracker [50]. However, the underlying assumption that corresponding pixels have the same color may be violated, *e.g.*, inevitable lighting changes and deformation in future frames, thereby hindering the model from using longer temporal cues. Recently, Jabri *et al.* [15] formulate a multi-step association that connects corresponding patches at every timestep between t and $t+k$ in a form of Markov chain. At each step, the patch-level affinity links all patches of two adjacent frames, preserving all possible correspondences in the video. The learning process thus benefits from a longer-range clip with all intermediate views available. Unfortunately, finding an "optimal" correspondence is not easy due

to the struggling matching between similar instances where image patches only capture a very *narrow* view of them.

Hence, we identify another key ingredient for better query-target matching — *seeing broader* — which is ignored in existing methods. Let us take a closer look of an example shown in Figure 1. How do human track the right dog tail across frames, and avoid being confused by the left dog tail from the similar instance? (i) *Seeing longer*: how the shape of the tail changes over time is indeed a crucial cue, and it can be utilized in the multi-step association [15]. (ii) *Seeing broader*: it is easier to discriminate the dog tails from a broader view by considering neighboring information around them, such as the dog body features, as well as the dog-person interaction. However, it cannot be achieved by straightforwardly enlarging the patch size, as the detailed structures or features will be missed.

In this paper, we propose to learn correspondence by *seeing both broader and longer* via a graph-based framework. We represent the video as a joint space-time graph where nodes are grid patches and edges are two type of relations, *i.e*., neighbor relations and similarity relations. The big graph can thus be decomposed into two sub-graphs. (i) **Neighbor Relation Graph**: we start by constructing a small graph for each node, which is linked to intra-frame nodes that are located in a sliding neighborhood. Initialized with the topological prior, the edges learn to guide the aggregation of neighboring node representations to the central node. The updated node representation thus captures a broader view of the neighborhood. (ii) **Similarity Graph**: we then connect inter-frame nodes with the pairwise similarity, under the updated node representations. All these edges form a multi-step association for a long-range clip.

Given the joint graph, the prediction of the long-range correspondence can be computed as a path (a combination of similarity-based edges) along the graph. To induce supervision, we adopt palindrome sequences [15] for training, which provide the walker with a target, *i.e*., returning to the start point. In contrast to the prior work [15], our path-level constraint provides contrastive learning signals from both the temporal views and neighboring views, leading to more reliable matching among similar instances. Moreover, we perform a random but attentive walk on the large graph by wisely dropping the "common-fate" [51] nodes according to the pixel discrepancy of each node, encouraging the model to focus on more informative node pairs. Below, we summarize the major contributions of this work.

- First, we design a joint video graph that models neighbor relations in space and similarity relations in time for visual correspondence learning.
- Second, we formulate the contrastive learning as a random but attentive walk on the graph to learn discriminative representations from seeing both temporal and neighboring views of instances.
- Third, our method outperforms state-of-the-art self-supervised approaches on a variety of visual tasks, *e.g*., object, part propagation, and pose tracking. It also surpasses some task-specific fully supervised algorithms.

2. Related Works

Self-supervised Representation Learning. Learning visual representations from unlabeled images or videos has been widely explored in many *pretext* tasks, including future prediction [39, 27], frame sorting [23, 30], motion estimation [1, 42], and audio analysis [34, 19]. These methods learn good feature representations that can generalize well to multiple tasks by further fine-tuning on a small set of labeled samples. The key idea in them is to utilize the inherent information inside images or videos as the supervisory signals. For example, corresponding pairs can be constructed by augmentation of the same instance [52, 3]. However, manually augmenting still images may not always be in correct correspondence. Recent works in *contrastive learning* [33, 6, 11, 10] explore the supervisory signals for similarity learning by choosing pairs that are close in space [11, 6, 2] or time [10, 33, 37]. In contrast, we implicitly determine which pairs to be closer by their neighbor relations in space and similarity relations in time.

Self-supervised Correspondence Learning. Recent approaches focus on learning correspondence from unlabeled videos in a self-supervised manner. The key idea of TimeCycle [50] is to train a deterministic patch tracker to find the correspondence of a query patch by tracking forward and backward in the video. Likewise, UVC [24] and ContrastCorr [47] adopt the patch tracker to obtain object-level correspondences. But they also explore fine-grained correspondences by learning a pixel-wise affinity with colorization. The difference is that Wang *et al*. [47] combines the intra-video transformation [24] with inter-video transformation to form contrastive pairs. Besides, CorrFlow [22] and MAST [21] use feature maps with higher resolution ($2\times$) than others and yield impressive results. Recently, Jabri *et al*. [15] formulate the correspondence as a contrastive random walk, allowing associations between patches that may have significant differences in appearance. Despite the success of these methods, many of them still struggle with the overwhelming noisy or negative samples when performing query-target matching. Our approach tackles this issue by introducing neighboring views to the matching pairs for contrastive learning, which allows us to lean implicit associations between central representation and its neighbor.

Video Graphs. Representing video as graphs can usually capture the spatial-temporal relationships in videos [41, 49,

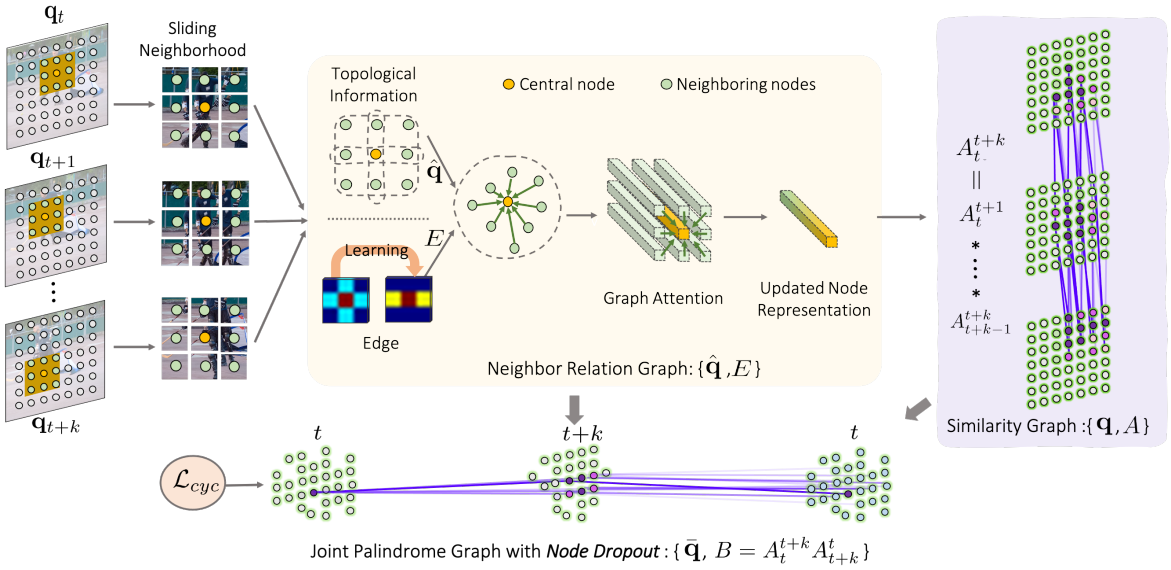

Figure 2. Schematic illustration of our joint space-time graph for correspondence learning. Specifically, we have two sub-graphs that associate grid patches (nodes) with different relations. (i) Neighbor Relation Graph $\{\hat{\mathbf{q}}, E\}$: it connects a central node to its neighbors $\hat{\mathbf{q}}$ with edge E initialized with topological prior, by which the neighboring embeddings can be aggregated to the center in a learnable manner. (ii) Similarity Graph $\{\mathbf{q}, A\}$: it links inter-frame nodes $\mathbf{q}$ with pair-wise similarity affinities A (in updated representation space) to form a multi-step association on a long-range sequence. Furthermore, we employ the node dropout technique and transfer sequence as palindrome to upgrade graph to $\{\bar{\mathbf{q}}, B\}$, in which we perform a random but attentive walk to find the correspondence based on contrastive learning.

36, 15]. The key of video graph is to form the image patches as nodes and link them with edges. One popular direction is to model the object-object interactions by connecting objects which overlap in space or close in time. They have been widely applied to video classification [49], detection [36], or visual relationship reasoning [41], by combining with Conditional Random Fields (CRF) [20] or Graph Convolutional Networks (GCN) [18]. Recently, some works start to model how the states of the same object change in time by connecting inter-frame nodes that have similar appearance or semantically related [15, 49]. Leveraging the similarity relations, the task of representation learning can be induced by propagating the node identity in a graph. To better learn instance discrimination with cross-attention between nodes, we go one step further by modelling the neighbor relations of intra-frame nodes, which allow us to learn latent associations between central node and its neighbors across space and time.

3. Methods

We propose to represent the video as a joint space-time graph for learning temporal correspondence. As shown in Figure 2, nodes are frame patches sampled in a grid, and edges contain two type of connections: neighbor relations between intra-frame nodes, and visual similarity between inter-frame nodes. Based on these two relations, the big graph can be decomposed into two sub-graphs, including neighbor relation graph and similarity graph, aiming at capturing the *broader* and *longer* views, respectively. Next, we perform reasoning on the graph to find latent correspondences for contrastive learning. Our learning process can be interpreted as a random but attentive walk on the graph by automatically dropping out some "common-fate" nodes. The construction details of two sub-graphs and learning procedures are successively described in this section.

3.1. Neighbor Relation Graph

Given a video sequence $\mathbf{I}$, we denote a set of nodes $\mathbf{q}_t$ that correspond to N overlapped patches sampled in a grid from frame $\mathbf{I}_t$. In general, each node in $\mathbf{q}_t$ will be mapped to a l_2-normalized d-dimensional embedding using an encoder ϕ, where d is the channel number. The embedding captures a very local view of an object sometimes, although necessary for learning tiny structures, it easily induces ambiguity into the query-target matching. Intuitively, the neighboring nodes provide the central node with a broader view of an object or interactions with other objects, which are beneficial to find its temporal correspondence. Motivated by this, we build a neighbor relation graph that reinforces node embeddings by relating the information from the neighbors guided by the general topology. The edge E is solely established between node i and its neighbors rather than all other nodes [28]. The sum of all edge values connected to node i is normalized to be 1 by a softmax function. If node i and j are spatially closer or more correlated in semantics,

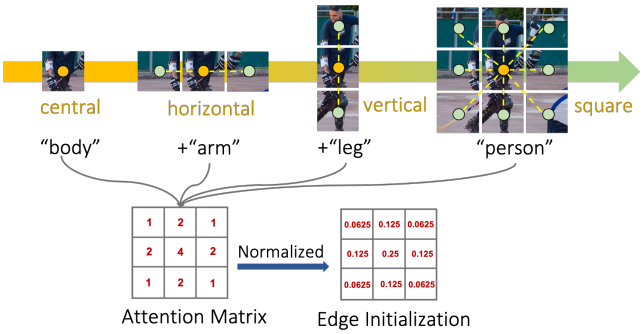

Figure 3. **Encoding topological information** in the neighbor relation graph by edge initialization. Three structures (*i.e.*, horizontal, vertical, square) associate the central node to broader representations variously. We adopt the normalized attention matrix as the initial value of edges to encode the topological prior.

then E_{ij} should be higher. We denote the neighbor relation graph as $G_r = \{\hat{\mathbf{q}}, E\}$ where $\hat{\mathbf{q}}$ is the set of n neighbors.

Sliding neighborhood. We construct G_r by considering a friendly neighborhood [13, 17] as a small grid (*e.g.*, a 3×3 grid yielding 9 neighbors), and model the neighbor relations for the central node. A larger neighborhood is not necessary because farther nodes are more likely to induce noise (see performance degradation in Figure 6 (a)). For different nodes in $\mathbf{q}_t$, we consider them as the center and determine corresponding neighborhoods in a sliding window manner. This lead to a shared edge $E \in \mathbb{R}^{N \times n}$ across N nodes in $\mathbf{q}_t$, *i.e.*, $E_{1j} = E_{2j} = ... = E_{Nj}$, where $j \in \{1, 2, ..., n\}$.

Encoding topological prior. To model the neighbor relation, we first initialize E by explicitly encoding the topological information. In a neighborhood (see Figure 3), we generally have three types of topology with regard to the central node based on the spatial proximity, including vertical, horizontal, and square structures. Each of them may composite central element into a higher-level entity in the feature space with neighboring views. For example, a "body center" can be extended with semantics of "arm", "legs", or "person" using horizontal, vertical, or square topology. We do have other cases that object-object interactions are captured by the neighbors, which also provide an crucial cue in modelling neighbor relations. To this regard, we generate a normalized attention matrix based on the number of node occurrences in different topology, and employ the matrix to initialize edge E. The topological prior can therefore be encoded to guide the following learning of E.

Graph attention. Advanced by the topological information, the graph captures the *relational importance* of neighboring nodes, in other words, the degree of importance of each of the neighbors contributing to the central node. We then explore a graph attention mechanism to augment the representations of the central node i by aggregating messages from its neighboring nodes:

$$f(\mathbf{q}_t^i) = \sum_{j=0}^{n} \texttt{softmax}(E_{ij}) \cdot \phi(\mathbf{q}_t^j), \tag{1}$$

where $\mathbf{q}_t^j$ is the j-th node in $\mathbf{q}_t$ and $f(\mathbf{q}_t^i)$ is the updated embedding of node $\mathbf{q}_t^i$, which provides weighted neighboring semantics, while preserving the original feature patterns. Different from the channel-level feature aggregation with GCN [49, 18], our graph attention performs a node-level feature simulation that treats each node embedding as a whole. We show this mechanism benefits the contrastive learning in Section 3.3. More importantly, E is learnable via back propagation, by which a more general neighbor relation can be modeled during training.

3.2. Similarity Graph

After considering the intra-frame node relations, we link the visually corresponding nodes in the adjacent frames by a similarity-based affinity. One general option for the pairwise similarity function is a dot-production between two feature embeddings: $F(\phi(q_1), \phi(q_2)) = \phi(q_1)^\top \phi(q_2)$. Following recent similarity learning methods [15, 24, 47], we employ a row-wise softmax function to the similarity function with temperature τ to obtain a non-negative affinity matrix between inter-frame node embeddings updated by G_r:

$$A_t^{t+1}(i,u) = \frac{\exp(F(f(\mathbf{q}_t^i),\ f(\mathbf{q}_{t+1}^u))/\tau)}{\sum_{l=1}^{N} \exp(F(f(\mathbf{q}_t^i),\ f(\mathbf{q}_{t+1}^l))/\tau)}. \tag{2}$$

The affinity in Equation (2) places weights on all possible edges between nodes at t and $t+1$, with higher possibility indicating that the paired patches are more similar. Given such connections, we can already construct a simple similarity graph between two adjacent frames in the video. However, the short temporal dynamics within two frames provides very limited views of objects. We therefore link all the inter-frame nodes across a video with a length of T, and formulate the graph path as a Markov chain of edges, following the idea in [15]:

$$A_t^{t+T} = \prod_{i=0}^{\mathrm{T}-1} A_{t+i}^{t+i+1}. \tag{3}$$

Here, we can denote the similarity graph as $G_s = \{\mathbf{q}, A\}$ where $\mathbf{q}$ represents all inter-frame nodes in a video and A is a chain of edges described in Equation (3).

3.3. Attentive Walk on Joint Space-Time Graph

Our goal is to learn temporal correspondence in the joint space-time graph without human annotations. Akin to prior arts that explore cycle-consistency in time [50,

15], we adopt the *palindrome* sequence in the form of $\{I_t, ..., I_{t+T}, ..., I_t\}$ for training, where the target of the query node should be its original position. Following the idea of [15], we build our cycle-consistent loss as:

$$\mathcal{L}_{cyc}(G|G = \{\mathbf{q}, B\}) = \mathcal{L}_{CE}(B, I), \tag{4}$$

where G is a joint palindrome graph with edges $B = A_t^{t+T} A_{t+T}^t$, and I is the target position generated according to the location of the first frame node, *e.g.*, the ground truth of the i-th node is i.

Node dropout. Compared with *things* — countable objects such as people and animals, one problem of learning correspondence in graphs is matching *stuff* — large regions of similar textures or materials, such as sky and land (Figure 4). The "common-fate" [51] nodes in the stuff have strong affinity to all other nodes in the related segment of neighboring frames, making it hard and somewhat impossible to walk back to the original position during training. To address this issue, we propose a node dropout strategy based on the pixel discrepancy of a node, given that the "common-fate" nodes always contain very similar context. Specifically, we start by retrieving the pixel embeddings $p \in \mathbb{R}^{d \times hw}$ for each node using the encoder ϕ, where hw is the downsampled spatial size. Next, we calculate the self-similarity among pixel embeddings via a dot-production: $S = p^\top p$. We define the pixel discrepancy of a node as:

$$\delta = 1 - \frac{1}{(hw)(hw)} \sum_{i=0}^{hw} \sum_{j=0}^{hw} S_{ij}, \tag{5}$$

where we convert the self-similarity to the reverse side, so that the higher value of δ denotes higher discrepancy among pixels. We then set a threshold of δ to drop those uninformative nodes whose pixel discrepancy is lower than it. The proposed strategy is superior to the random dropout technique in [15] by exclusively tackling "common-fate" nodes. Our final training objective is:

$$\mathcal{L}_{cyc}(\bar{G}|\bar{G} = \{\bar{\mathbf{q}}, B^k\}) = \sum_{k=1}^{\mathrm{T}} \mathcal{L}_{CE}(B^k, I), \tag{6}$$

where $\bar{\mathbf{q}}$ is the remained nodes after node dropout, and $B^k = A_t^{t+k} A_{t+k}^t$. We optimize all sub-cycles in the graph with the clip length k varying from 1 to T. In this regard, we are allowed to perform a random but attentive walk on the graph, forcing the model to discriminate informative node pairs by the hedging of ambiguous matching.

Contrastive learning with *extra* positive pairs. We can consider our model as a chain of contrastive learning problem, which is guided by a "one-hop" cycle-consistency constraint. Basically, a strong edge creates a one-to-one alignment, *i.e.*, one positive pair. However, after aggregating neighboring information via Equation (1), we can interpret

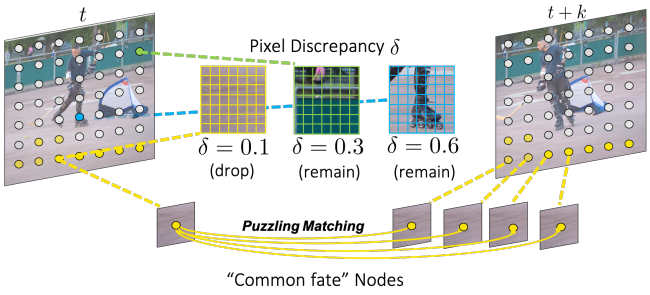

Figure 4. **Node dropout** to avoid "common-fate" nodes that puzzle the correspondence learning. We introduce a thresholding measurement based on the pixel discrepancy δ.

one edge as a *latent* many-to-many alignment in the feature space, involving *extra* positive pairs for contrastive learning apart from center-center pairs. For example, center-neighbor pairs — a node at timestep t and one of its corresponding neighbors at timestep $t+1$, or even neighbor-neighbor pairs. Taking the hand-arm pair as an example, since they are physically connected as neighbors in most cases, when learning the correspondence of "hand", our model may push its embedding closer to that of "arm" as well. Recall that we learn E in Equation (1) via the objective (6). The learned E encourages the model to find more reliable center-center or center-neighbor pairs for contrastive learning, which generate better node representations in return for E to model the general neighbor relations. We believe this is the main reason that our model learns more discriminative representations of instances.

4. Experiments

We extensively evaluate our learned representations on various visual correspondence tasks: video object propagation, human part propagation, and pose keypoint tracking. We first conduct the comparison with the state-of-the-art self-supervised algorithms for visual correspondence learning, including TimeCycle [50],CorrFlow [22], MAST [21], UVC [24], ContrastCorr [47], and VideoWalk [15]. We then compare our model with pre-trained features from representation learning methods, including MoCo [11], a self-supervised contrastive learning method based on images; VINCE [10], an extension of MoCo to videos; ImageNet [12], a strong supervised method where the model is pre-trained on ImageNet. All above methods use ResNet-18 [12] as the backbone. Besides, we also compare with some fully supervised algorithms designed for the specific tasks. At last, we provide in-depth ablation studies.

4.1. Implementation

Encoder. For fair comparisons, we also adopt ResNet-18 [12] as the encoder ϕ by reducing the stride of last two residual blocks (`res3` and `res4`) to 1. We add a linear pro-

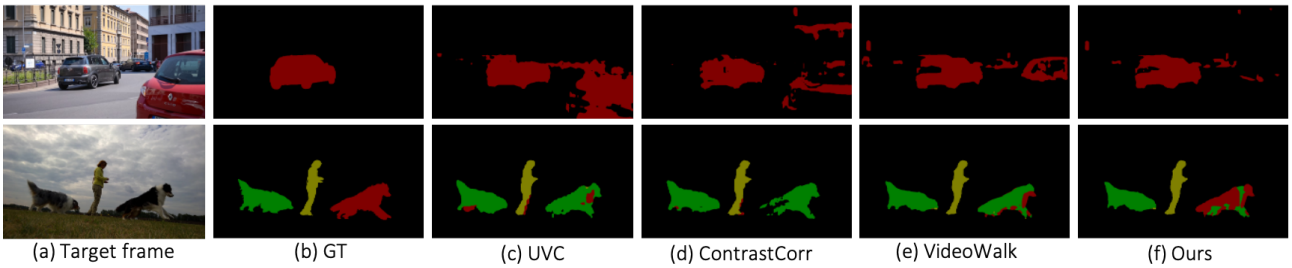

Figure 5. Qualitative comparisons with other self-supervised methods on DAVIS 2017 dataset. (a) Target frame. (b) Ground-truth of target frame. (c) Results of UVC [24]. (d) Results of ContrastCorr [47]. (e) Results of VideoWalk [15]. (f) Our results.

Table 1. **Video object propagation** results on DAVIS 2017 dataset. We show results of state-of-the-art self-supervised methods and some supervised approaches in comparison of our method. *Train Data* indicates dataset(s) used for pre-training, including: I = ImageNet [12], K = Kinetics400 [5], C = CoCo [25], D = DAVIS 2017 [35], P = PASCAL-VOC [8], Y = YouTube-VOS [53], O = OxUvA [43], V = VLOG [9], T = TrackingNet [31]. *Resolution* indicates whether feature map for correspondence matching is of a higher ($2\times$) resolution.

Method	Supervised	Backbone	Train Data	Resolution	$\mathcal{J}\&\mathcal{F}_m$	$\mathcal{J}_m$	$\mathcal{J}_r$	$\mathcal{F}_m$	$\mathcal{F}_r$
MoCo [11]		ResNet-18	I	$1\times$	60.8	58.6	68.7	63.1	72.7
VINCE [10]			K		60.4	57.9	66.2	62.8	71.5
CorrFlow [22]		ResNet-18	O	$2\times$	50.3	48.4	53.2	52.2	56.0
MAST [21]			O		63.7	61.2	73.2	66.3	78.3
MAST [21]			Y		65.5	63.3	73.2	67.6	77.7
TimeCycle [50]		ResNet-18	V	$1\times$	48.7	46.4	50.0	50.0	48.0
UVC [24]			K		60.9	59.3	68.8	62.7	70.9
ContrastCorr [47]			T		63.0	60.5	70.6	65.5	73.0
VideoWalk [15]			K		67.6	64.8	76.1	70.2	82.1
Ours		ResNet-18	K	$1\times$	**68.7**	**65.8**	**77.7**	**71.6**	**84.3**
ImageNet [12]	✓	ResNet-18	I	$1\times$	62.9	60.6	69.9	65.2	73.8
SiamMask [48]	✓	ResNet-50	I/C/Y		56.4	54.3	62.8	58.5	67.5
OSVOS [4]	✓	VGG-16	I/D		60.3	56.6	63.8	63.9	73.8
OnAVOS [44]	✓	ResNet-38	I/C/P/D		65.4	61.6	67.4	69.1	75.4
OSVOS-S [29]	✓	VGG-16	I/P/D		68.0	64.7	74.2	71.3	80.7

jection after average pooling to generate a 128-dimensional node embedding. Pixel-wise embeddings of each node are created by the same projection without average pooling for node dropout.

Training. We train ϕ using the unlabeled videos from Kinetics400 [5] dataset with the Adam optimizer. We set the temperature τ as 0.05 in Equation 2. Akin to [15], for each 256×256 frame, we sample 64×64 patches in a 7×7 grid, resulting in 49 nodes per frame. Without extra specification, our sliding neighborhood is a 3×3 grid, involving 9 neighboring nodes, and the length of training sequences is 10. To see sufficient samples, we first train the model without node dropout for 5 epochs using a learning rate of 1×10^{-4}. Next, we set $\delta = 0.2$ for node dropout and train the model for another 15 epochs with a learning rate of 1×10^{-5}. All experiments are conducted on 4 NVIDIA Titan Xp GPUs.

Inference. All evaluation tasks can be considered as video label propagation, which is to predict the labels of each pixels in the target frame given only the labels in the first frame (*i.e.*, the source). For fair comparisons, we use the same label propagation strategy and testing protocols as [15] for all tasks. In brief, labels L_t are propagated as $L_t = K_t^s L_s$, where L_s is the source labels and K_t^s is the top-k transitions between source and target frames (k is 10 for all tasks). To provide temporal context, the last m frames are also used for propagation (m is 20, 4, and 7 for DAVIS, VIP and JHMDB tasks respectively). To avoid noise from pixels that are far away in space, the query pixels are restricted by a *local* attention mask with a radius r (r is 5 for JHMDB, and 12 for all other tasks). We use the output of `res3` as feature representations to calculate affinities for label propagation, and τ is set as 0.05 for consistency with the training.

4.2. Video Object Propagation on DAVIS 2017

We evaluate our model on a popular benchmark of semi-supervised video object segmentation, *i.e.*, DAVIS 2017 [35], which provides the semantic mask of multiple objects in the first frame. For fair comparisons with prior works [15, 47, 24, 50], we test the model on images with the resolution of 480p. We report the mean (m) and recall (r) of Jaccard index $\mathcal{J}$ (IoU) and contour alignment $\mathcal{F}$, detailed in Table 1. Figure 5 and Figure 7 (a) show the propagated object masks. Specifically, our approach attains improvements over MoCo [11] and VINCE [10], indicat-

Table 2. **Part segmentation and Pose tracking** results on VIP and JHMDB datasets, respectively. We compare our model with self-supervised and strong supervised methods. *Sup* indicates it is a supervised method or not.

Method	Sup	Pose		Part
		PCK@0.1	PCK@0.2	mIoU
TimeCycle [50]		57.3	78.1	28.9
UVC [24]		58.6	79.6	34.1
ContrastCorr [47]		61.1	80.8	37.4
VideoWalk [15]		59.3	84.9	38.6
Ours		**61.4**	**85.3**	**40.2**
ImageNet [12]	✓	53.8	74.6	31.9
ATEN [54]	✓	-	-	37.9
Thin-Slicing Net [38]	✓	68.7	92.1	-

ing that it is better to choose temporal views for contrastive learning in videos rather than data augmentation of a single frame. Our method also performs favourably against the self-supervised methods from unlabeled video, and even without relying on a higher-resolution feature map used in CorrFlow and MAST [22, 21] or other modules such as the patch localizer designed in UVC and ContrastCorr [24, 47]. Our method achieves consistent improvements over the state-of-the-art method VideoWalk [15] across all the evaluation metrics. Surprisingly, our method can outperform many supervised methods [48, 4, 44, 29] with specific architectures for video object segmentation.

Furthermore, we show that by modelling neighbor relations during training, our model exhibits superior discrimination capability for instance-level separation. As seen in Figure 5, both UVC [24] and ContrastCorr [47] fail to discriminate similar instances by learning pixel-level correspondence. Despite seeing more hard negative samples during training, the walker in [15] still gets confused to instances that are similar in color. In contrast, our model can tell the difference of small parts between similar instances, *e.g.*, the dog tails or the car windows, by inducing neighboring views for contrastive learning.

4.3. Human Part Propagation on VIP

We evaluate our method on part segmentation task on the Video Instance Parsing (VIP) benchmark [54], which involves propagating 20 parts of human (*e.g.*, arms and leg), requiring more precise matching than DAVIS. We use the same settings as Jabri *et al*. [15], and resize the video frames to 560×560. For the semantic part propagation task, we evaluate performance via the mean IoU metric. As seen in Table 2, our model outperforms existing self-supervised methods, *e.g.*, by 1.6% and 2.8% mIoU, compared to VideoWalk [15] and ContrastiveCorr [47], respectively. We also surpass the fully supervised method ATEN [54] that is specifically designed for this dataset using training labels. Figure 7 (b) shows samples of semantic part propagation results. Interestingly, our model correctly propagates each part mask onto similar instances (dancers in the first example) no matter when they are close or far from the camera.

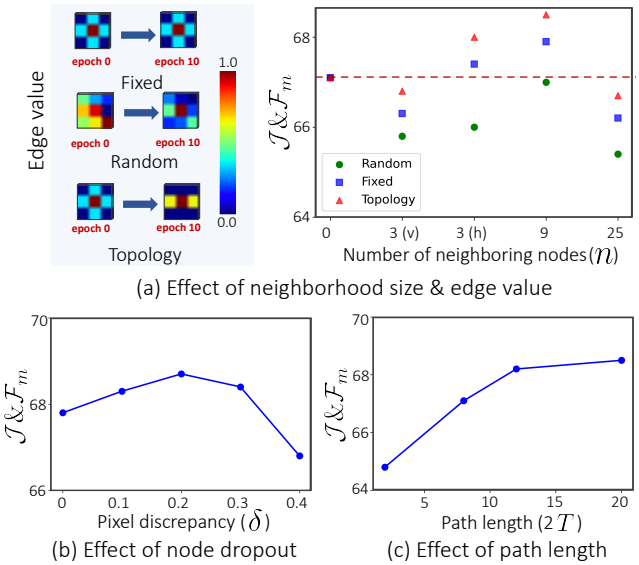

Figure 6. Ablation studies of our method on DAVIS 2017 benchmark. (a) Effect of neighborhood size and edge value. (b) Effect of node dropout. (c) Effect of training path length.

4.4. Pose Keypoint Tracking on JHMDB

We consider the pose tracking task on JHMDB benchmark [16], which involves 15 keypoints. Follow the evaluation protocol of [24, 15], we test the model on 320×320px images. We adopt the possibility of correct keypoints [38] (PCK) as the evaluation metric, which measures the percentage of keypoints close to ground-truth under different thresholds. We show quantitative evaluations against others in Table 2, and qualitative results in Figure 7 (c). Our model achieves consistent improvements over existing self-supervised approaches on this challenging task that requires precise fine-grained matching. Notably, our model achieves even 10.7% better in PCK@0.2 than the ImageNet [12] baseline trained with classification labels.

4.5. Analytical Ablation Studies on DAVIS 2017

Neighborhood size. We investigate how necessary to relate more neighboring nodes for a broader view, by constructing neighbor relation graph with dimension of 3×1, 1×3, 3×3, 5×5 — resulting in $n = 3, 3, 9, 25$ nodes, respectively. In Figure 6 (a), we find that 9 neighboring nodes can peak the performance on DAVIS. Further increasing the neighborhood size ($n = 25$) is likely to induce noisy cues from farther nodes, resulting in even worse results than the baseline that does not consider neighbor relations. Interestingly, the interactions with horizontally-connected nodes are more beneficial to learn discriminative representations than the vertical ones (see 3 (h) *vs*. 3 (v)).

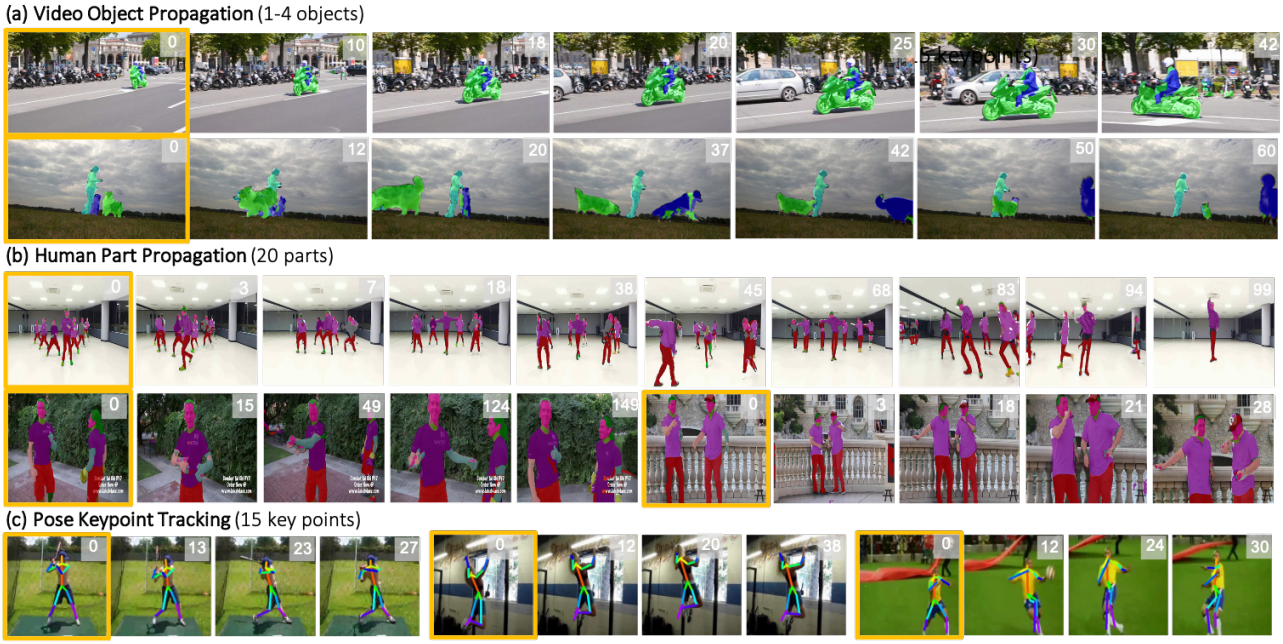

Figure 7. Propagation results of our model. (a) Video object propagation on DAVIS 2017 [35] dataset. (b) Human part propagation on VIP [54] dataset. (c) Pose keypoint tracking on JHMDB [16] dataset. The first frame is highlighted with a yellow outline with its label being provided. Without fine-tuning, our model achieves promising long-range label propagation on three visual tasks.

Variants of edge E. We also explore three variants of edge E in the neighbor relation graph in Figure 6 (a): (i) *Fixed*: fixed edge with topological information; (ii) *Random*: learnable edge with random initialization; (iii) *Topology*: learnable edge with topological initialization. In conclusion, encoding topology is essential for modelling neighbor relation of nodes. Further learning edge E at training time yields better results. We attribute this success by more general neighbor relations gained in learning procedures.

Node dropout. We evaluate the effect of node dropout by training our model with ranging values of $\delta \in [0, 0.4]$ at a step of 0.1. Higher δ means more nodes will be dropped based on their pixel discrepancy. In Figure 6 (b), we find that moderate node dropout (*i.e.*, $0.1 \sim 0.3$) boost the performance on DAVIS, with $\delta = 0.2$ peaking the result. It demonstrates that the technique can tackle "common-fate" nodes, helping the model focus on informative contents.

Path length. In Figure 6 (c), we explore the effect of path length during training. Using clips of length 2, 4, 6, 10, we obtain paths of length 4, 8, 12, 20 for training. We see that longer sequences can improve results on DAVIS. This observation is similar to previous work [15], as model can see longer views of instances for contrastive learning.

Component analysis. We analyze the key components of our model in Table 3. The similarity graph G_s alone yields unsatisfactory results due to the puzzling negative samples. By modelling neighbor relations in G_r, the performance is greatly improved by 2.2% in $\mathcal{J}\&\mathcal{F}_m$. Further using node dropout on the joint graph peaks the performance.

Table 3. **Component analysis** of our model on DAVIS benchmark.

G_s	G_r	Node Dropout	$\mathcal{J}\&\mathcal{F}_m$
✓			65.6
✓	✓		67.8 (+2.2%)
✓	✓	✓	68.7 (+3.1%)

5. Conclusion

In this work, we present a novel self-supervised approach for learning correspondence from unlabeled videos. Our key idea is to explore the structures and dynamics of objects from both neighboring and temporal views. To achieve this, we learn to walk on a joint space-time graph that connects nodes with neighbor relations and similarity relations. The superiority of our learned representation is demonstrated on three video propagation tasks. Without fine-tuning, our method outperforms the state-of-the-art self-supervised methods, as well as some strong fully supervised models that are designed for specific tasks. In the future, we plan to handle those "extremely similar instances" — whose correspondences are hard to find based on visual similarity — by leveraging the motion patterns [40] or depth information [55] from large-scale unlabeled videos.

Acknowledgement

This work was supported by Hong Kong Research Grants Council with Project No. CUHK 14201620.

References

[1] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *Proceedings of the IEEE international conference on computer vision*, pages 37–45, 2015.

[2] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *arXiv preprint arXiv:1906.00910*, 2019.

[3] Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise. In *International Conference on Machine Learning*, pages 517–526. PMLR, 2017.

[4] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 221–230, 2017.

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[7] 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 *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.

[8] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015.

[9] David F Fouhey, Wei-cheng Kuo, Alexei A Efros, and Jitendra Malik. From lifestyle vlogs to everyday interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4991–5000, 2018.

[10] 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.

[11] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3464–3473, 2019.

[14] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017.

[15] Allan Jabri, Andrew Owens, and Alexei A. Efros. Space-time correspondence as a contrastive random walk. In *Advances in Neural Information Processing Systems*, 2020.

[16] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 3192–3199, 2013.

[17] Dongkwan Kim and Alice Oh. How to find your friendly neighborhood: Graph attention design with self-supervision. In *International Conference on Learning Representations*, 2021.

[18] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[19] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 7774–7785, 2018.

[20] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[21] Zihang Lai, Erika Lu, and Weidi Xie. Mast: A memory-augmented self-supervised tracker. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6479–6488, 2020.

[22] Zihang Lai and Weidi Xie. Self-supervised learning for video correspondence flow. *arXiv preprint arXiv:1905.00875*, 2019.

[23] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 667–676, 2017.

[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 *Advances in Neural Information Processing Systems*, pages 318–328, 2019.

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[26] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):978–994, 2010.

[27] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*, 2016.

[28] Jiaqi Ma, Bo Chang, Xuefei Zhang, and Qiaozhu Mei. Copulagnn: Towards integrating representational and correlational roles of graphs in graph neural networks. *arXiv preprint arXiv:2010.02089*, 2020.

[29] 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 pattern analysis and machine intelligence*, 41(6):1515–1530, 2018.

[30] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016.

[31] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 300–317, 2018.

[32] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9226–9235, 2019.

[33] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[34] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 631–648, 2018.

[35] 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.

[36] Xufeng Qian, Yueting Zhuang, Yimeng Li, Shaoning Xiao, Shiliang Pu, and Jun Xiao. Video relation detection with spatio-temporal graph. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 84–93, 2019.

[37] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1134–1141. IEEE, 2018.

[38] Jie Song, Limin Wang, Luc Van Gool, and Otmar Hilliges. Thin-slicing network: A deep structured model for pose estimation in videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4220–4229, 2017.

[39] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852. PMLR, 2015.

[40] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning motion patterns in videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3386–3394, 2017.

[41] Yao-Hung Hubert Tsai, Santosh Divvala, Louis-Philippe Morency, Ruslan Salakhutdinov, and Ali Farhadi. Video relationship reasoning using gated spatio-temporal energy graph. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10424–10433, 2019.

[42] Hsiao-Yu Fish Tung, Hsiao-Wei Tung, Ersin Yumer, and Katerina Fragkiadaki. Self-supervised learning of motion capture. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5242–5252, 2017.

[43] Jack Valmadre, Luca Bertinetto, Joao F Henriques, Ran Tao, Andrea Vedaldi, Arnold WM Smeulders, Philip HS Torr, and Efstratios Gavves. Long-term tracking in the wild: A benchmark. In *Proceedings of the European conference on computer vision (ECCV)*, pages 670–685, 2018.

[44] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. *arXiv preprint arXiv:1706.09364*, 2017.

[45] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *Proceedings of the European conference on computer vision (ECCV)*, pages 391–408, 2018.

[46] Ning Wang, Yibing Song, Chao Ma, Wengang Zhou, Wei Liu, and Houqiang Li. Unsupervised deep tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1308–1317, 2019.

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

[48] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1328–1338, 2019.

[49] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *Proceedings of the European conference on computer vision (ECCV)*, pages 399–417, 2018.

[50] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2566–2576, 2019.

[51] Max Wertheimer. Laws of organization in perceptual forms. 1938.

[52] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

[53] 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.

[54] Qixian Zhou, Xiaodan Liang, Ke Gong, and Liang Lin. Adaptive temporal encoding network for video instance-level human parsing. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1527–1535, 2018.

[55] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017.