Exposing the Self-Supervised Space-Time Correspondence Learning via Graph Kernels

Zheyun Qin[1], **Xiankai Lu**[1*], **Xiushan Nie**[2], **Yilong Yin**[1*], **Jianbing Shen**[3]
[1] School of Software, Shandong University
[2] School of Computer Science and Technology, Shandong Jianzhu University
[3] SKL-IOTSC, CIS, University of Macau
zheyunqin@gmail.com, carrierlxk@gmail.com

Abstract

Self-supervised space-time correspondence learning is emerging as a promising way of leveraging unlabeled video. Currently, most methods adapt contrastive learning with mining negative samples or reconstruction adapted from the image domain, which requires dense affinity across multiple frames or optical flow constraints. Moreover, video correspondence predictive models require mining more inherent properties in videos, such as structural information. In this work, we propose the VideoHiGraph, a space-time correspondence framework based on a learnable graph kernel. Concerning the video as the spatial-temporal graph, the learning objectives of VideoHiGraph are emanated in a self-supervised manner for predicting unobserved hidden graphs via graph kernel manner. We learn a representation of the temporal coherence across frames in which pairwise similarity defines the structured hidden graph, such that a biased random walk graph kernel along the sub-graph can predict long-range correspondence. Then, we learn a refined representation across frames on the node-level via a dense graph kernel. The self-supervision of the model training is formed by the structural and temporal consistency of the graph. VideoHiGraph achieves superior performance and demonstrates its robustness across the benchmark of label propagation tasks involving objects, semantic parts, keypoints, and instances. Our algorithm implementations have been made publicly available at https://github.com/zyqin19/VideoHiGraph.

Introduction

Self-supervised learning (SSL) paradigm seeks to mine supervisory information from the unsupervised data and design functional pretext learning tasks. Guided by the idea of SSL, self-supervised representation learning based on still images has made a flurry of advances, especially using contrastive learning (Chen et al. 2020; Xie, Wang, and Ji 2020), yet this has rarely translated to the field of video understanding (Wang et al. 2021a).

Video is often understood as a variant in that the image extends the time dimension, which limits the progress of video understanding (Feichtenhofer et al. 2019; Jabri, Owens, and Efros 2020; Araslanov, Schaub-Meyer, and Roth 2021)—extending contrastive learning to videos bringing a huge computation burden and unsatisfying performance, especially for label propagation tasks that require fine-grained localization ability in each frame. One potential solution resorts to mining temporal correspondence (*i.e.*, cycle-consistency) across the video as temporal supervision (Vondrick et al. 2017; Wang et al. 2018; Lai and Xie 2019; Wang, Jabri, and Efros 2019; Wang et al. 2021b; Xu and Wang 2021; Wang et al. 2021a). On account of the hypothesis about intrinsic slow feature changing properties (Turner and Sahani 2007; Wiskott and Sejnowski 2002; Lu et al. 2020a), these methods have achieved promising performance across several label propagation tasks, such as video object segmentation (Lu et al. Jun. 2019,A; Wang et al. 2022), pose tracking (Li et al. 2019), and semantic part propagation (Zhou et al. 2018). Nevertheless, these methods depend on repeatedly applying complex greedy tracking and adjacent frame sampling training over time that may be trapped in local optimality.

Jabri, Owens, and Efros (2020) recently argued that video temporal correspondences are implicit supervision, and exploit directed graph view to represent the video. This way, the temporal correspondence can be comprehended as patch-level similarity learning. While this is a promising direction, the proposed method needs dense affinity computation among multiple frames and neglects the underlying structural information in the video (Narayanan and Mitter 2010).

In this paper, we develop an SSL framework via hidden graph prediction for space-time correspondence learning of video data, termed as VideoHiGraph (See Fig. 1). Firstly, we construct a spatial-temporal graph to describe a video, where each image patch is formulated as a node and the similarity among the intra-frame and inter-frame as the edge. A good correspondence for the label propagation task implies that the nodes inside each frame should automatically cluster into their respective semantic part (sub-graph); meanwhile, sub-graphs in adjacent frames share a similar motion trend. With this spirit, we employ graph kernels to capture spatial and temporal structure in the video and take the structural consistency to guide correspondence learning.

We first employ hidden graphs to incorporate reliable nodes in the spatial graph and build the target-related sub-graph automatically. Then, we connect the sub-graphs across different frames via a random walk graph kernel. Considering the missing annotations in the videos, we additionally utilize the idea of temporal consistency as node-level supervision.

*Corresponding authors.

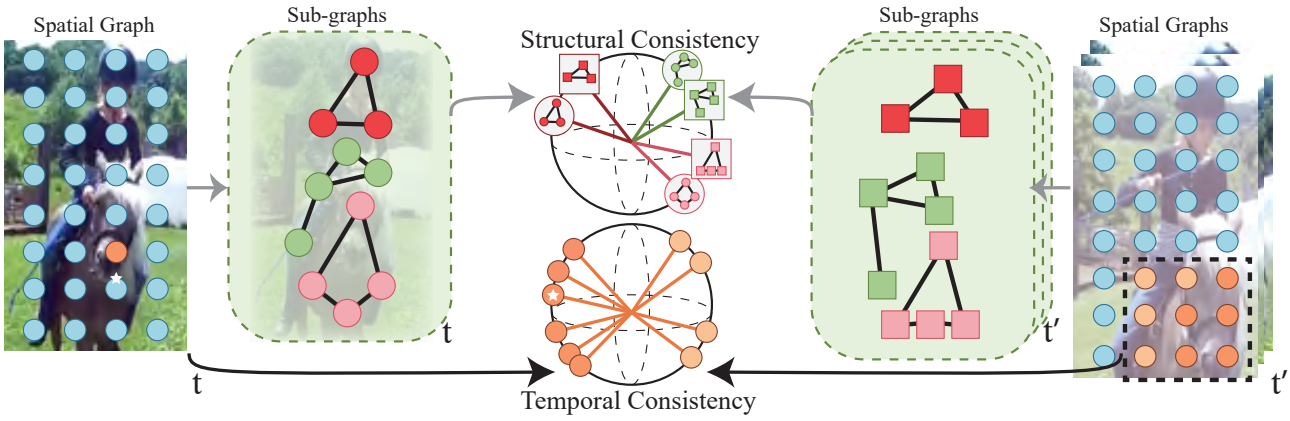

Figure 1: Illustration of the main idea. We learn video space-time correspondence in an unsupervised manner by (i) the structural consistency and (ii) the temporal consistency. Through the structural consistency, we can learn the sub-graph representation of the spatial graph (the green rectangular box) and then capture the spatio-temporal structure of the video. The temporal consistency is treated as supervised to provide good node-level representation.

In this way, our solution can suppress the adverse impact of ambiguous matches. The effectiveness of VideoHiGraph on graph-level and node-level representation learning is demonstrated on various label propagation tasks.

To summarise, our contributions are as follows:

- We propose a new self-supervised space-time correspondence learning framework, VideoHiGraph, which is a complementary scheme to exploit video supervision from the graph and node-level perspectives.
- We involve the biased graph kernel to produce structural representations during model learning. This kernel allows our model to mine more free supervision via learnable hidden graphs without any heuristics.
- We demonstrate the model's high transparency on video correspondence tasks, *i.e.*, video object segmentation, semantic part propagation, pose tracking and video instance segmentation. On these downstream tasks, our model even outperforms task-specific fully-supervised ones.

Related Work

Temporal Correspondence. In video understanding and analysis, temporal correspondence is a crucial factor in learning motion states and inevitably moving trajectories of objects. Traditional approaches tend to formulate a self-supervised learning paradigm as a colorization proxy task (Vondrick et al. 2017; Lai and Xie 2019; Li et al. 2019; Wang, Jabri, and Efros 2019; Lai, Lu, and Xie 2020) by recovering the color correspondence of pixels. However, the underlying assumption of color consistency (*i.e.*, the corresponding pixels have similar colors) is invalid for lighting change or deformation. Other methods (Janai et al. 2018; Yin and Shi 2018; Meister, Hur, and Roth 2018; Wang et al. 2019; Wang, Jabri, and Efros 2019; Jabri, Owens, and Efros 2020; Son 2022) use contrastive cycle-consistency in time to supervise model training.

Recent approaches (Wang, Zhou, and Li 2021; Lu et al. 2020c; Xu and Wang 2021; Zhao, Jin, and Heng 2021; Xu and Wang 2021; Lu et al. 2021; Li et al. 2022) take a step further with surrogate video classes, pre-aligned patch pairs, video frame-level similarity, or neighboring views to exploit the inter-video context. While these methods are impressive, they lack a crucial element for robust correspondence: structure correspondence of inter-video. In response, we employ the hidden graph better to capture spatial and temporal structure among the video to learn correspondence.

Self-supervised Representation Learning. For static image-based self-supervised learning, the solutions based on contrastive learning reduce the performance gap to a supervised counterpart (Chen et al. 2020; Grill et al. 2020; He et al. 2020). For video sources, several previous studies (Asano, Rupprecht, and Vedaldi 2020; Feichtenhofer et al. 2021) have found that adding additional temporally-invariant or temporally-persistent constraints to these action recognition methods can bring considerable performance improvements. In addition, some work (Fernando et al. 2017; Dave et al. 2022) is explicitly tailored to learning spatial-temporal representations of video recognition and understanding tasks. However, these efforts are restricted by typically computational complexity in label propagation tasks, as they do not learn **fine-grained** correspondences in the context, at the object or instance level.

Self-supervised Graph Representation Learning. Graph representation learning method models the distribution of nodes with connectivity in the graph. Specifically, graph embedding and graph kernel are the mainstreams of existing self-supervised graph representation methods. The graph embedding methods are mainly based on stochastic (*i.e.*, random walk) heuristics (Perozzi, Al-Rfou, and Skiena 2014; Grover and Leskovec 2016; Hamilton, Ying, and Leskovec 2017) while sampling negatives randomly. However, these methods overemphasize proximity information at the expense of structural information (Hassani and Khasahmadi 2020) and do not take full advantage of node features (You, Ying, and Leskovec 2019). The graph kernel methods aspect (Borgwardt and

Kriegel 2005; Shervashidze et al. 2009, 2011; Chen, Jacob, and Mairal 2020) decomposes graphs into substructure (including random walks, shortest paths, subtrees, graphlets, *etc.*) and measures the pairwise similarity among sub-graphs by the kernel function. Nevertheless, these methods require artificially designed measures to measure the similarity between substructures. Instructively, RWNN (Nikolentzos and Vazirgiannis 2020) compares a spatio-temporal graph (input) against learnable graphs with the random walk kernel for learning graph representations, which focus on structure. Inspired by RWNN (Nikolentzos and Vazirgiannis 2020), we exploit a biased random walk kernel to characterize the local neighborhoods and macro-view structure in videos.

Method

Preliminaries of Graph Kernel

General Formulations of Graph Kernel. Graph kernels give rise to well-defined function spaces to identify a compact sub-graph structure and possess rules of composition that guide how input graphs can be built from simpler subgraph (*i.e.*, hidden graphs) (Lei et al. 2017; Jagarlapudi and Jawanpuria 2020; Nikolentzos and Vazirgiannis 2020). Regarding unlabeled video as a graph, we compare the input graphs against several learnable hidden graphs to learn the sub-graph structure. We formulate this as an optimization task that maximizes the similarity between the distributions of the input graph and the hidden graphs with graph kernels in a random walk manner.

Formally, we define two graphs as $\boldsymbol{G}_1 = (\boldsymbol{V}_1, \boldsymbol{E}_1)$ and $\boldsymbol{G}_2 = (\boldsymbol{V}_2, \boldsymbol{E}_2)$ with $|\boldsymbol{V}_1| = n$ and $|\boldsymbol{V}_2| = m$, and the direct product as $\boldsymbol{G}_\times = (\boldsymbol{V}_\times, \boldsymbol{E}_\times)$ with $|\boldsymbol{V}_\times| = nm$ that is a graph over pairs of nodes from $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ with nodes set $\boldsymbol{V}_\times = \{(v_1, v_2) : v_1 \in \boldsymbol{V}_1 \wedge v_2 \in \boldsymbol{V}_2\}$ and edges set $\boldsymbol{E}_\times = \{\{(v_1, v_2), (u_1, u_2)\} : \{v_1, u_1\} \in \boldsymbol{E}_1 \wedge \{v_2, u_2\} \in \boldsymbol{E}_2\}$.

In concrete, the random walk graph kernel performs random walks on two graphs $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ separately, quantifying the similarity between pairwise graphs by the number of matched walks. We perform simultaneous random walks within two graphs, which can be formalized as a random walk on $\boldsymbol{G}_\times$ in an elegant way to find the correspondence between hidden graphs (Gärtner, Flach, and Wrobel 2003; Nikolentzos and Vazirgiannis 2020).

$$k_\times(\boldsymbol{G}_1, \boldsymbol{G}_2) = \sum_{i,j=1}^{|\boldsymbol{V}_\times|} \left[\sum_{p=0}^{P} (\lambda \boldsymbol{A}_\times)^p \right]_{ij}, \tag{1}$$

where $\boldsymbol{A}_\times \in \mathbb{R}^{nm \times nm}$ is the adjacency matrix of $\boldsymbol{G}_\times$ (Xu et al. 2019), λ is the positive, real-valued bias, and P is the step of the walk.

Furthermore, we extend the scalar-node in G_1, G_2 with learnable vectors $\boldsymbol{F}_1 \in \mathbb{R}^{n \times d}$, $\boldsymbol{F}_2 \in \mathbb{R}^{m \times d}$ to better depict the node relationship. Then, let $\boldsymbol{F}_\times = \boldsymbol{F}_1 \boldsymbol{F}_2^\top \in \mathbb{R}^{n \times m}$ be the similarity between the two graphs' nodes. The graph kernel in Eq. 1 can be further expanded to consider the node similarity with bias, and the P-step random walk kernel with

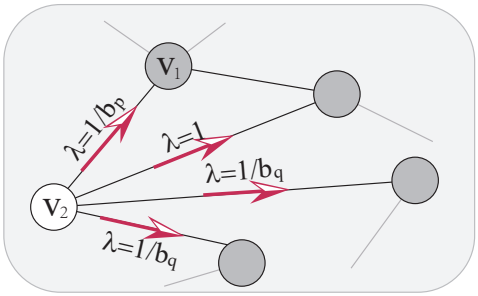

Figure 2: Illustration of the bias of graph kernel. The walker has walked from node v_1 to node v_2, now evaluating the next walk's possible step (red arrow).

$\mathbf{f} = \mathrm{vec}(\boldsymbol{F}_\times) \in \mathbb{R}^{nm}$ is defined as:

$$\begin{aligned} k_\times(\boldsymbol{G}_1, \boldsymbol{G}_2) &= \sum_{i,j=1}^{|\boldsymbol{V}_\times|} \mathbf{f}_i \mathbf{f}_j \left[\sum_{p=0}^{P} \lambda_p \boldsymbol{A}_\times^p \right]_{ij} \\ &= \sum_{p=0}^{P} \mathbf{f}^\top \lambda_p \boldsymbol{A}_\times^p \mathbf{f}. \end{aligned} \tag{2}$$

The Bias of Graph Kernel. Aimed to capture the entire graph's structural information efficiently, we hope to take homophily and structural equivalence (Henderson et al. 2011) into consideration when exploring diverse local neighborhoods in the process of random walking. To this end, we further developed a flexible biased random walk to obtain the learnable hidden graph, as shown in Fig. 2.

To be concrete, for bias in each step (λ in Eq. 2), we introduce a second-order random walk guided by two learnable parameters b_p and b_q (Grover and Leskovec 2016). After walking from node v_1 to v_2 and now residing at node v_2, the probability of the next walk to node v_3 is considered as:

$$\lambda_{b_p, b_q}(v_1, v_3) = \begin{cases} 1/b_p, & \text{if } d_{v_1, v_3} = 0 \\ 1, & \text{if } d_{v_1, v_3} = 1 \\ 1/b_q, & \text{if } d_{v_1, v_3} = 2 \end{cases}, \tag{3}$$

where $d_{v_1, v_3} = 1$ or 2 denote that v_3 is the one-hop or two-hop neighbor of v_1, $d_{v_1, v_3} = 0$ indicates that the walker has returned from node v_2 to node v_1.

Framework

In the context of video correspondence learning, we present our VideoHiGraph using the proposed biased graph kernel. Let v_{clip} be an input video clip with T frames, which is constructed as a spatial-temporal graph. G_t is the spatial graph whose nodes represent the patch features extracted from t^{th} frame and mapped by an encoder ϕ, and nodes in adjacent frames share an edge. As shown in Fig. 3, our method learns both graph-level and node-level correspondence.

Graph-Level Correspondence Learning For graph-level correspondence learning, we first extract the spatial structure *i.e.*, sub-graphs, of the spatial graph in each frame. Then, the temporal graph is constructed based on cross-frame matching sub-graphs. Building upon a spatial-temporal graph to

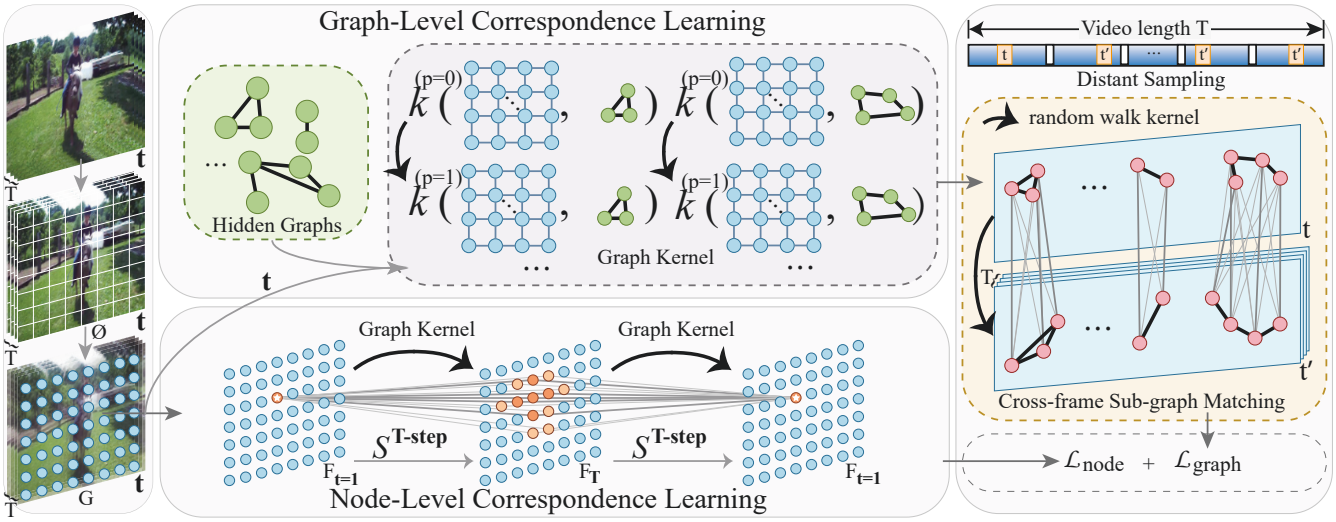

Figure 3: The proposed model for correspondence learning on graph and node levels. We construct the spatial graph based on grid patches (nodes) of t frame. (i) Graph-level learns sub-graph representing the spatial structure of t frame based on biased graph kernel with hidden graphs. Next, we build the temporal graph to learn long-range correspondence by matching the same sub-graphs in t and t' frames. (ii) Node-level links all cross-frame nodes to construct temporal graph to provide self-supervised signals based on graph kernels.

describe the video, we can encode structural information or multiple-nodes relationship into the representations.

Concretely, we initialize H learnable hidden graphs (*i.e.*, the adjacency matrix and node features are learnable) as sub-graphs and compare the spatial graph $\boldsymbol{G}_t$ against these hidden graphs $\mathcal{G}_t = \{\mathcal{G}_t^{(1)}, \cdots, \mathcal{G}_t^{(h)}, \cdots, \mathcal{G}_t^{(H)}\}, h \in \{1, \cdots, H\}$. Following the above preliminaries, the structural features of the sub-graph can be obtained as:

$$\begin{aligned} \mathcal{K}_t^{(h)} &= f(cat[k_\times^1(\boldsymbol{G}_t, \mathcal{G}_t^{(h)}) \cdot F_t^{(h)}, \ldots, k_\times^P(\boldsymbol{G}_t, \mathcal{G}_t^{(h)}) \cdot F_t^{(h)}]), \\ \mathcal{K}_t &= \{\mathcal{K}_t^{(1)}, \ldots, \mathcal{K}_t^{(h)}, \ldots, \mathcal{K}_t^{(H)}\}, \end{aligned} \tag{4}$$

where '$\cdot$' denotes the dot product, $f(\cdot)$ and $cat[\cdot,\cdot]$ mean linear projection and concatenation operation, $k_\times^p(\boldsymbol{G}_t, \mathcal{G}_t^{(h)})$ is the p^{th}-step graph kernel value, $p \in \{1, \cdots, P\}$ (See Eq. 2), $F_t^{(h)} \in \mathbb{R}^{m \times d}$ is the node feature of $\mathcal{G}_t^{(h)}$, m and d denote the number and feature dimension of nodes, respectively.

Furthermore, we employ cross-frame matching to associate the sub-graphs (Zeng et al. 2019; Brasó and Leal-Taixé 2020). We first sample two sub-graph matching features $\mathcal{K}_t^{(h)}, \mathcal{K}_{t'}^{(h)} \in \mathbb{R}^{m \times d}$ in different frames t, t' by distant sampling strategies (Xu and Wang 2021) and match these features of two graphs with the matching matrix $\boldsymbol{M} \in \{0,1\}^{m \times m}$. The optimization of $\boldsymbol{M}$ is achieved by maximizing the following object function (Jagarlapudi and Jawanpuria 2020):

$$\max_{\boldsymbol{M}} \mathtt{Tr}(\boldsymbol{M}\boldsymbol{K}_\times), \tag{5}$$

where $\boldsymbol{K}_\times = \mathcal{K}_{t'}^{(h)} \mathcal{K}_t^{(h)\top}$, $\mathtt{Tr}(\cdot)$ denotes the trace of a matrix. To solve Eq. 5, one can relax $\boldsymbol{M}$ to be an element of the *transportation polytope* (Asano, Rupprecht, and Vedaldi 2020; Cuturi 2013):

$$\begin{aligned} &\max_{\boldsymbol{M}} \mathtt{Tr}(\boldsymbol{M}^\top \boldsymbol{K}_\times) + \kappa h(\boldsymbol{M}), \\ &s.t. \quad \boldsymbol{M} \in \mathbb{R}_+^{m \times m}, \ \boldsymbol{M}^\top \mathbf{1}^m = \mathbf{1}^m, \boldsymbol{M}\mathbf{1}^m = \mathbf{1}^m, \end{aligned} \tag{6}$$

where $\mathbf{1}^m$ denotes the vector of all ones of m dimensions. $h(\boldsymbol{M}) = -\sum_{i,j} \boldsymbol{M}_{i,j} \log \boldsymbol{M}_{i,j}$ is entropy as the extra regularization term to reduce the randomness of node matching, controlled by hyperparameters $\kappa > 0$. We use a small value of κ to ensure that each node of $\mathcal{K}_t^{(h)}$ is assigned to one and only one node of $\mathcal{K}_{t'}^{(h)}$, and vice versa. With the soft assignment relaxation, the solver of Eq. 6 can be given as (Cuturi 2013):

$$\boldsymbol{M} = \mathtt{diag}(\boldsymbol{u}') \exp\left(\frac{\boldsymbol{K}_\times}{\kappa}\right) \mathtt{diag}(\boldsymbol{v}'), \tag{7}$$

where $\boldsymbol{u}' \in \mathbb{R}^m$ and $\boldsymbol{v}' \in \mathbb{R}^m$ are renormalization vectors following *Sinkhorn-Knopp iteration* (Cuturi 2013).

With the matching probability matrix $\boldsymbol{M}$, the i^{th} node of $\mathcal{K}_t^{(h)}$ is assigned to the j^{th} node of $\mathcal{K}_{t'}^{(h)}$, where $j = \mathrm{argmax}_{i'}\{\boldsymbol{M}_{i,i'}\}_{i'=1}^m$. Furthermore, $\mathcal{P}_{t,i}^{(h)} = \{\mathcal{K}_{t',j}^{(h)}\}_{t'=1}^T$ denotes the best matching node set of $\mathcal{K}_{t,i}^{(h)}$ across the T frames, and the other node set is $\mathcal{U}_{t,i}^{(h)} = \{\mathcal{K}_{t',z}^{(h)}\}_{t',z=1}^{T,m} / \mathcal{P}_{t,i}^{(h)}$. Briefly speaking, objects in each frame is modeled as multiple sub-graphs. Here, we use the InfoNCE loss (Van den Oord, Li, and Vinyals 2018) to preserve the structure consistency:

$$\mathcal{L}_{graph} = \sum_{h,t,i=1}^{H,T,m} \sum_{\mathbf{p} \in \mathcal{P}_{t,i}^h} \log \frac{-\exp(\mathcal{K}_{t,i}^{(h)} \cdot \mathbf{p}/\tau_p)}{\exp(\mathcal{K}_{t,i}^{(h)} \cdot \mathbf{p}/\tau_p) + \sum_{\mathbf{u} \in \mathcal{U}_{t,i}^h} \exp(\mathcal{K}_{t,i}^{(h)} \cdot \mathbf{u}/\tau_p)}, \tag{8}$$

where τ_p is a hyper-parameter.

Node-Level Correspondence Learning Building on graph-level correspondence, we employ the temporal consistency on the node-level temporal graph to refine the space-time correspondence. Inspired by CRW (Jabri, Owens, and Efros 2020), we construct a temporal graph in that the adjacent frame nodes share a directed edge, and the pairwise similarity of nodes denotes weight. The node-level correspondence is obtained by performing a forward-backward cycle on the temporal graph, taking the weight of edges as the transition probability.

More precisely, let the dense node feature maps of $\boldsymbol{G}_t$ and $\boldsymbol{G}_{t+1}$ be $\boldsymbol{F}_t$ and $\boldsymbol{F}_{t+1}$. The non-negative affinities matrix is computed over edges departing from each node as:

$$\boldsymbol{S}_t^{t+1}(i,j) = \frac{\exp(< \boldsymbol{F}_t^i, \boldsymbol{F}_{t+1}^j > /\tau_a)}{\sum_{l=1}^{N} \exp(< \boldsymbol{F}_t^i, \boldsymbol{F}_{t+1}^l > /\tau_a)}. \tag{9}$$

For a video clip with T frames in total, we perform a T-step random walk on the temporal graph:

$$\bar{\boldsymbol{S}}_1^T = \prod_{i=0}^{T-2} \text{softmax}(\boldsymbol{S}_{i+1}^{i+2}). \tag{10}$$

With the matching matrix $\bar{\boldsymbol{S}}_1^T$, similar to Eq. 8, the node-level objective function is minimized as follows:

$$\begin{gathered} \mathcal{L}_{node} = \mathcal{L}_{cross}(\boldsymbol{R}, \boldsymbol{I}), \\ \text{where } \boldsymbol{R} = \bar{\boldsymbol{S}}_1^T \bar{\boldsymbol{S}}_T^1 \Rightarrow \textit{forward-backward cycle}, \end{gathered} \tag{11}$$

where $\mathcal{L}_{cross}$ is cross entropy loss function, and $\boldsymbol{I}$ is the target position index generated according to the location of the t frame node, *e.g.*, the ground truth of the i^{th} node is i.

Total Loss. Finally, we minimize the hidden-graph and node-level objective with the weight β:

$$\mathcal{L}_{total} = \mathcal{L}_{graph} + \beta \mathcal{L}_{node}. \tag{12}$$

Implementation

We generate node embedding (*i.e.*, image patch) with the encoder ϕ-ResNet-18 (He et al. 2016). We reduce the down-sampling factor to 1/8. For each frame, we sampling overlapping 32×32 patches in an 8×8 grid as nodes, and then fed them into ϕ to extract node features. Our training videos come from Kinetics400 (Carreira and Zisserman 2017) without using any annotation labels. $T = 18$ is the length of the input video clip. We set $T/2 = 9$ for the distant sampling strategy of the graph-level correspondence, and node-level continuous sampling (Xu and Wang 2021) is set with a fixed frame interval of 2 from the starting frame.

The hyper-parameters are empirically set to: $H = 5$, $P = 3$, $\kappa = 0.03$, $\tau_p = 0.1$, $\tau_a = 0.07$, $\beta = 0.5$. We use the Adam solver to optimize the loss function. The learning rate is set to 1×10^{-4}, and the weight decay is 1×10^{-6}.

Experiments

Evaluation of Learned Representation

Following previous works (Wang, Jabri, and Efros 2019; Wang et al. 2021b), the evaluation does not involve additional finetune. We evaluated our VideoHiGraph on two types of fine-grained video correspondence tasks: *one-shot* paradigm (Lai and Xie 2019; Wang et al. 2021b) and *zero-shot* paradigm (Lei, Xing, and Chen 2020; Lu et al. 2020b; Qin et al. 2021). The *one-shot* paradigm is treated as label propagation that propagates the initial label of the first frame is to subsequent frames according to the dense correspondence learned. Meanwhile, the *zero-shot* paradigm casts the evaluation as a temporal association that needs to track and segment all objects automatically. The first task involves object segmentation, semantic parts tracking, and human pose tracking, while the latter involves instance association.

Video Object Segmentation In Table 1, We first evaluate our method based on the popular video object segmentation benchmark DAVIS-2017 (Pont-Tuset et al. 2017). VideoHiGraph is considerably superior to all current SSL methods, leading the second-best CLTC (Jeon et al. 2021) and the third-best CRW (Jabri, Owens, and Efros 2020) by 0.2 and 2.2 in terms of $\mathcal{J}\&\mathcal{F}_m$, respectively. Remarkably, CLTC used task-specific model weights and architectures for DAVIS. It suggests that our graph kernel-based model captures the dynamic structure of the video better than the densely connected approach. Meanwhile, our SSL approach, trained without pixel-wise manual annotations, achieves competitive performance compared to some famous supervised models (*i.e.*, OSVOS (Caelles et al. 2017) and FEELVOS (Voigtlaender et al. 2019)). This result indicates that structural and temporal consistency provides a good supervise signal and dramatically affects the correspondence quality.

In Table 2, we examine our SSL method on another VOS benchmark, *i.e.*, YouTube-VOS (Xu et al. 2018), again confirming the advantage of our approach among all the above SSL methods and partially famous supervision methods. The performance reported and the visualization of the learned hidden graphs in Fig. 4 confirm the superiority and validity of our approach to exploring video structure, allowing our model to learn more robust representations.

Video Part Segmentation and Pose Keypoint Tracking We next evaluate our VideoHiGraph on the Video Instance Parsing (VIP) benchmark (Zhou et al. 2018) for the body part propagation task and the JHMDB benchmark (Jhuang et al. 2013) for the pose keypoint tracking task. VideoHiGraph consistently outperforms all above SSL methods (See Table 3) on semantic-level (mIoU), instance-level (AP), and keypoint-level (PCK) parsing, especially the main counterpart: CRW and CLTC. For the VIP, the performance is attributed to the method's ability that learns well cross-instance differentiation and intra-instance invariance. For the JHMDB, the success indicates that the model learns robust representations from sufficiently negative samples to learn sub-graph structure.

Video Instance Segmentation We finally examine the model performance on the video instance segmentation benchmark - YouTube-VIS (Yang, Fan, and Xu 2019) for experimental completeness. Our VideoHiGraph exhibits superior performance on the instance-level association (See Table 4), and outperforms the seminal tracking-by-detection work, *i.e.*, MaskTrack R-CNN (Yang, Fan, and Xu 2019), and the bottom-up method, *i.e.*, STEm-Seg (Athar et al. 2020).

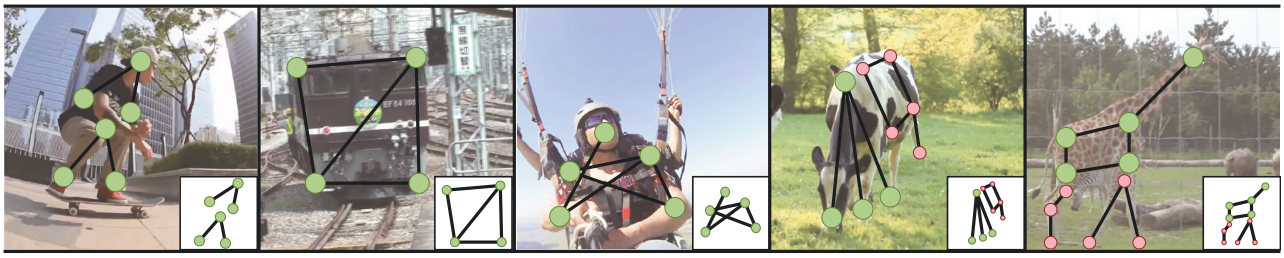

Figure 4: Examples of hidden graphs learned (lower right corner) from the graph-level correspondence learning component and some target-related sub-graphs of the YouTube-VOS.

Method	Backbone	Sup.	Dataset (size)	$\mathcal{J}\&\mathcal{F}_{\mathrm{m}}$	$\mathcal{J}_{\mathrm{m}}$	$\mathcal{J}_{\mathrm{r}}$	$\mathcal{F}_{\mathrm{m}}$	$\mathcal{F}_{\mathrm{r}}$
Colorization (Vondrick et al. 2017)	ResNet-18	✗	Kinetics (800 hours)	34.0	34.6	34.1	32.7	26.8
CorrFlow (Lai and Xie 2019)	ResNet-18	✗	OxUvA (14 hours)	50.3	48.4	53.2	52.2	56.0
UVC (Li et al. 2019)	ResNet-50	✗	C/Kinetics (800 hours)	60.9	59.3	68.8	62.7	70.9
MAST (Lai, Lu, and Xie 2020)	ResNet-18	✗	OxUvA (14 hours)	63.7	61.2	73.2	66.3	78.3
VFS (Xu and Wang 2021)	ResNet-18	✗	Kinetics (800 hours)	66.7	64.0	-	69.4	-
CRW (Jabri, Owens, and Efros 2020)	ResNet-18	✗	Kinetics (800 hours)	68.3	65.5	78.6	71.0	82.9
NRG (Zhao, Jin, and Heng 2021)	ResNet-18	✗	Kinetics (800 hours)	68.7	65.8	77.7	71.6	84.3
CLTC (Jeon et al. 2021)	ResNet-18	✗	YT (5.58 hours)	70.3	67.9	78.2	72.6	83.7
Ours	ResNet-18	✗	Kinetics (800 hours)	**70.5**	**67.9**	**80.7**	**73.1**	**83.8**
ResNet (He et al. 2016)	ResNet-18	✓	I (1.28M, -)	62.9	60.6	69.9	65.2	73.8
OSVOS (Caelles et al. 2017)	VGG-16	✓	I/D (1.28M, 10k)	60.3	56.6	63.8	63.9	73.8
FEELVOS (Voigtlaender et al. 2019)	Xception-65	✓	I/C/D/YT (1.28M, 663k)	71.5	69.1	79.1	74.0	83.8
STM (Oh et al. 2019)	ResNet-50	✓	I/D/YT (1.28M, 164k)	81.8	79.2	88.7	84.3	91.8

Table 1: Quantitative comparisons for video object segmentation on DAVIS-2017. "Sup." indicates whether it is a supervised model. For convenience, we simplified the representation of some datasets, *e.g.*, ImageNet(I), COCO(C), DAVIS(D), PASCAL-VOC(P), YouTube-VOS(YT).

Method	Backbone	Sup.	Overall	$\mathcal{J}_{Seen}\uparrow$	$\mathcal{F}_{Seen}\uparrow$	$\mathcal{J}_{Unseen}\uparrow$	$\mathcal{F}_{Unseen}\uparrow$
Colorization (Vondrick et al. 2017)	ResNet-18	✗	38.9	43.1	38.6	36.6	37.4
CorrFlow (Lai and Xie 2019)	ResNet-18	✗	46.6	50.6	46.6	43.8	45.6
MAST (Lai, Lu, and Xie 2020)	ResNet-18	✗	64.2	63.9	64.9	60.3	67.7
CLTC (Jeon et al. 2021)	ResNet-18	✗	67.3	66.2	67.9	63.2	71.7
Ours	ResNet-18	✗	**67.8**	**66.3**	**68.1**	**64.5**	**72.1**
OSVOS (Caelles et al. 2017)	VGG-16	✓	58.8	59.8	60.5	54.2	60.7
PreMVOS (Luiten, Voigtlaender, and Leibe 2018)	ResNet101	✓	66.9	71.4	75.9	56.5	63.7
STM (Oh et al. 2019)	ResNet-50	✓	79.4	79.7	84.2	72.8	80.9

Table 2: Quantitative comparisons for video object segmentation on YouTube-VOS with "Seen" and "Unseen" classes.

Note that our method with a lightweight backbone shows substantial competitive performance without annotations.

Further Analysis

Hidden Graphs. We first study the influence of our core differentiable biased graph kernel on performance. As shown in Table 5 (left), we study more variants by replacing the proposed biased graph kernel method with (1) the shortest path kernel (SP) (Borgwardt and Kriegel 2005), (2) the graphlet kernel (GR) (Shervashidze et al. 2009), and (3) the Weisfeiler-Lehman subtree kernel (WL) (Shervashidze et al. 2011).

Due to the learning of sub-graph structure, our differentiable biased graph kernel can recognize the fundamental properties of sub-graphs that are indistinguishable in other graph kernels (Kriege et al. 2018) (See Fig. 4), such as triangle-freeness (*i.e.*, a graph does not contain a cycle with three nodes). Benefiting from taking the homophily and structural equivalence into account, we explore a complementary solution and achieve better results than WL, confirming the superiority of our proposed biased graph kernel. Meanwhile,

Method	Backbone	Sup.	VIP		JHMDB	
			mIoU↑	AP ↑	PCK@0.1 ↑	PCK@0.2 ↑
TimeCycle (Wang, Jabri, and Efros 2019)	ResNet-50	✗	28.9	15.6	57.3	78.1
UVC (Li et al. 2019)	ResNet-50	✗	34.1	17.7	58.6	79.6
CRW (Jabri, Owens, and Efros 2020)	ResNet-18	✗	38.6	-	59.3	84.9
CLTC (Jeon et al. 2021)	ResNet-18	✗	37.8	19.1	60.5	82.3
Ours	ResNet-18	✗	**39.6**	**20.9**	**61.7**	**85.2**
ResNet (He et al. 2016)	ResNet-18	✓	31.9	12.6	53.8	74.6
ATEN (Zhou et al. 2018)	ResNet-50	✓	37.9	24.1	-	-
TSN (Song et al. 2017)	CPM (Wei et al. 2016)	✓	-	-	68.7	92.1

Table 3: Quantitative comparisons for part segmentation and pose tracking on VIP and JHMDB *val*, respectively.

Method	Backbone	Supervised	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
DeepSORT(Wojke, Bewley, and Paulus 2017)	ResNet-50	✓	26.1	42.9	26.1	27.8	31.3
SeqTracker(Yang, Fan, and Xu 2019)	ResNet-50	✓	27.5	45.7	28.7	29.7	32.5
MaskTrack R-CNN(Yang, Fan, and Xu 2019)	ResNet-50	✓	30.3	51.1	32.6	31.0	35.5
STEm-Seg(Athar et al. 2020)	ResNet-50	✓	30.6	50.7	33.5	31.6	37.0
Ours	ResNet-18	✗	**30.8**	**54.5**	**33.9**	**33.3**	**37.2**

Table 4: Quantitative comparisons for video instance segmentation on YouTube-VIS$_{21}$ *val*.

Kernel	$\mathcal{J}\&\mathcal{F}_{\text{m}}$
SP	67.7
GR	68.5
WL	68.9
Our	**70.5**

$\mathcal{L}_{\text{node}}$	$\mathcal{L}_{\text{graph}}$	Bias	$\mathcal{J}\&\mathcal{F}_{\text{m}}$
✓			68.4
✓	✓		69.9
✓	✓	✓	**70.5**

Table 5: Ablative studies of Graph Kernel Strategies (left) and Training Components $\mathcal{L}$ (right).

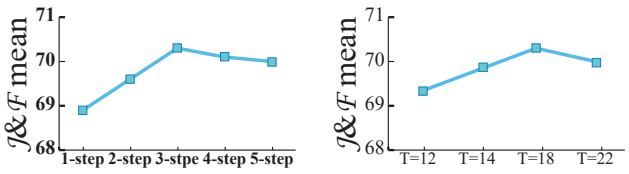

Figure 5: Ablative studies of Step Length P (left) and Sequence Length T (right).

we determine that the too-large step length P of the biased graph kernel hinders the performance as shown in Fig. 5 (left) that tends to over-complex the hidden graph learned.

Training Components. We next investigate the efficacy of essential components, *i.e.*, graph-level and node-level correspondence. In Table 5 (right), our parser modeling graph-level correspondence provides a substantial performance gain in $\mathcal{J}\&\mathcal{F}_m$, *i.e.*, +1.5 (see row 2), and +2.1 (see row 3), respectively. The graph-level component can distinguish similar (hard negative sample) or occluded objects from a structured perspective, and the node-level component complementary guarantees a confidence score in the self-supervised setting.

Biased Graph Kernels. We also validate the assumption of our biased graph kernels. Table 5 (right) is evident that biased graph kernels in exploring neighborhoods outperform non-biased graph kernels ($b_p = b_q = 1$, see row 2), giving us a 0.6 gain. The learnable parameters b_p and b_q can adaptively mine the right mix of homophily and structural equivalence to explore the graph's structure.

Sequence Length. Finally, we explore the contribution of the sequence length in the training phase. As shown in Fig. 5 (right), the best performance for our model can be obtained by setting the sequence length of the video clip to $T = 18$. We find that longer training sequences in conjunction with sampling strategy accelerated convergence and improved performance on the DAVIS task. Of course, an excessively long sequence length can increase the task's difficulty.

Conclusions

In this paper, we presented an SSL approach for temporal correspondence learning, VideoHiGraph, from unlabelled videos. Utilizing the prior of corresponding patches consistency, we train the hidden graph network to achieve temporal consistency based on the graph kernel. Then, cross-frame matching works for regularizing representation learning and improving correspondence inference. We argue that a unified framework is an appealing alternative to task-specific methods, and the effectiveness was thoroughly validated over various label propagation tasks. Despite the impressive results, we also expect a flurry of innovations along the direction of modalities consistency in areas such as video, text, audio, and more.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No. 61876098, 62176139, 62106128, 62176141), the Major basic research project of Shandong Natural Science Foundation (No. ZR2021ZD15), the Natural Science Foundation of Shandong Province (No. ZR2021QF001), the Young Elite Scientists Sponsorship Program by CAST (No. 2021QNRC001), the Open project of Key Laboratory of Artificial Intelligence, Ministry of Education, the Shandong Provincial Natural Science Foundation for Distinguished Young Scholars (ZR2021JQ26), the Taishan Scholar Project of Shandong Province (tsqn202103088).

References

Araslanov, N.; Schaub-Meyer, S.; and Roth, S. 2021. Dense Unsupervised Learning for Video Segmentation. In *NeurIPS*.

Asano, Y. M.; Rupprecht, C.; and Vedaldi, A. 2020. Self-labelling via simultaneous clustering and representation learning. In *ICLR*.

Athar, A.; Mahadevan, S.; Osep, A.; Leal-Taixé, L.; and Leibe, B. 2020. STEm-Seg: Spatio-temporal Embeddings for Instance Segmentation in Videos. In *ECCV*.

Borgwardt, K. M.; and Kriegel, H.-P. 2005. Shortest-path kernels on graphs. In *ICDM*.

Brasó, G.; and Leal-Taixé, L. 2020. Learning a neural solver for multiple object tracking. In *CVPR*.

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

Carreira, J.; and Zisserman, A. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*.

Chen, D.; Jacob, L.; and Mairal, J. 2020. Convolutional kernel networks for graph-structured data. In *ICML*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. E. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML*.

Cuturi, M. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*.

Dave, I.; Gupta, R.; Rizve, M. N.; and Shah, M. 2022. TCLR: Temporal contrastive learning for video representation. *Comput. Vis. Image Und.*, 219: 103–406.

Feichtenhofer, C.; Fan, H.; Malik, J.; and He, K. 2019. Slowfast networks for video recognition. In *ICCV*.

Feichtenhofer, C.; Fan, H.; Xiong, B.; Girshick, R. B.; and He, K. 2021. A Large-Scale Study on Unsupervised Spatiotemporal Representation Learning. In *CVPR*.

Fernando, B.; Bilen, H.; Gavves, E.; and Gould, S. 2017. Self-Supervised Video Representation Learning with Odd-One-Out Networks. In *CVPR*.

Gärtner, T.; Flach, P.; and Wrobel, S. 2003. On graph kernels: Hardness results and efficient alternatives. In *Learning theory and kernel machines*, 129–143. Springer.

Grill, J.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P. H.; Buchatskaya, E.; Doersch, C.; Pires, B. Á.; Guo, Z.; Azar, M. G.; Piot, B.; Kavukcuoglu, K.; Munos, R.; and Valko, M. 2020. Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning. In *NeurIPS*.

Grover, A.; and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *KDD*.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *NeurIPS*.

Hassani, K.; and Khasahmadi, A. H. 2020. Contrastive multi-view representation learning on graphs. In *ICML*.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. B. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR*.

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

Henderson, K.; Gallagher, B.; Li, L.; Akoglu, L.; Eliassi-Rad, T.; Tong, H.; and Faloutsos, C. 2011. It's who you know: graph mining using recursive structural features. In *KDD*.

Jabri, A.; Owens, A.; and Efros, A. 2020. Space-time correspondence as a contrastive random walk. In *NeurIPS*.

Jagarlapudi, S. N.; and Jawanpuria, P. K. 2020. Statistical Optimal Transport posed as Learning Kernel Embedding. In *NeurIPS*.

Janai, J.; Guney, F.; Ranjan, A.; Black, M.; and Geiger, A. 2018. Unsupervised learning of multi-frame optical flow with occlusions. In *ECCV*.

Jeon, S.; Min, D.; Kim, S.; and Sohn, K. 2021. Mining Better Samples for Contrastive Learning of Temporal Correspondence. In *CVPR*.

Jhuang, H.; Gall, J.; Zuffi, S.; Schmid, C.; and Black, M. J. 2013. Towards understanding action recognition. In *ICCV*.

Kriege, N. M.; Morris, C.; Rey, A.; and Sohler, C. 2018. A Property Testing Framework for the Theoretical Expressivity of Graph Kernels. In *IJCAI*.

Lai, Z.; Lu, E.; and Xie, W. 2020. MAST: A memory-augmented self-supervised tracker. In *CVPR*.

Lai, Z.; and Xie, W. 2019. Self-supervised learning for video correspondence flow. In *BMVC*.

Lei, C.; Xing, Y.; and Chen, Q. 2020. Blind video temporal consistency via deep video prior. In *NeurIPS*.

Lei, T.; Jin, W.; Barzilay, R.; and Jaakkola, T. 2017. Deriving neural architectures from sequence and graph kernels. In *ICML*.

Li, L.; Zhou, T.; Wang, W.; Yang, L.; Li, J.; and Yang, Y. 2022. Locality-Aware Inter-and Intra-Video Reconstruction for Self-Supervised Correspondence Learning. In *CVPR*.

Li, X.; Liu, S.; De Mello, S.; Wang, X.; Kautz, J.; and Yang, M.-H. 2019. Joint-task self-supervised learning for temporal correspondence. In *NeurIPS*.

Lu, X.; Ma, C.; Shen, J.; Yang, X.; Reid, I.; and Yang, M.-H. 2020a. Deep object tracking with shrinkage loss. *IEEE Trans. Pattern Anal. Mach. Intell.*

Lu, X.; Wang, W.; Ma, C.; Shen, J.; Shao, L.; and Porikli, F. Jun. 2019. See More, Know More: Unsupervised Video Object Segmentation With Co-Attention Siamese Networks. In *CVPR*.

Lu, X.; Wang, W.; Martin, D.; Zhou, T.; Shen, J.; and Luc, V. G. Aug. 2020. Video Object Segmentation with Episodic Graph Memory Networks. In *ECCV*.

Lu, X.; Wang, W.; Shen, J.; Crandall, D.; and Luo, J. 2020b. Zero-Shot Video Object Segmentation with Co-Attention Siamese Networks. *IEEE Trans. Pattern Anal. Mach. Intell.*

Lu, X.; Wang, W.; Shen, J.; Crandall, D. J.; and Van Gool, L. 2021. Segmenting objects from relational visual data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(11): 7885–7897.

Lu, X.; Wang, W.; Shen, J.; Tai, Y.-W.; Crandall, D. J.; and Hoi, S. C. 2020c. Learning video object segmentation from unlabeled videos. In *CVPR*.

Luiten, J.; Voigtlaender, P.; and Leibe, B. 2018. Premvos: Proposal-generation, refinement and merging for video object segmentation. In *ACCV*.

Meister, S.; Hur, J.; and Roth, S. 2018. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *AAAI*.

Narayanan, H.; and Mitter, S. 2010. Sample complexity of testing the manifold hypothesis. In *NeurIPS*.

Nikolentzos, G.; and Vazirgiannis, M. 2020. Random walk graph neural networks. In *NeurIPS*.

Oh, S. W.; Lee, J.-Y.; Xu, N.; and Kim, S. J. 2019. Video object segmentation using space-time memory networks. In *ICCV*.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *KDD*.

Pont-Tuset, J.; Perazzi, F.; Caelles, S.; Arbeláez, P.; Sorkine-Hornung, A.; and Van Gool, L. 2017. The 2017 DAVIS Challenge on Video Object Segmentation. *arXiv:1704.00675*.

Qin, Z.; Lu, X.; Nie, X.; Zhen, X.; and Yin, Y. 2021. Learning hierarchical embedding for video instance segmentation. In *ACM MM*.

Shervashidze, N.; Schweitzer, P.; Van Leeuwen, E. J.; Mehlhorn, K.; and Borgwardt, K. M. 2011. Weisfeiler-lehman graph kernels. *J. MACH. LEARN. RES.*, 12(9).

Shervashidze, N.; Vishwanathan, S.; Petri, T.; Mehlhorn, K.; and Borgwardt, K. 2009. Efficient graphlet kernels for large graph comparison. In *AISTATS*.

Son, J. 2022. Contrastive Learning for Space-Time Correspondence via Self-Cycle Consistency. In *CVPR*.

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

Turner, R.; and Sahani, M. 2007. A maximum-likelihood interpretation for slow feature analysis. *Neural computation*, 19(4): 1022–1038.

Van den Oord, A.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv e-prints*, arXiv–1807.

Voigtlaender, P.; Chai, Y.; Schroff, F.; Adam, H.; Leibe, B.; and Chen, L. 2019. FEELVOS: Fast End-To-End Embedding Learning for Video Object Segmentation. In *CVPR*.

Vondrick, C.; Shrivastava, A.; Fathi, A.; Guadarrama, S.; and Murphy, K. 2017. Tracking emerges by colorizing videos. In *ECCV*.

Wang, N.; Song, Y.; Ma, C.; Zhou, W.; Liu, W.; and Li, H. 2019. Unsupervised deep tracking. In *CVPR*.

Wang, N.; Zhou, W.; and Li, H. 2021. Contrastive transformation for self-supervised correspondence learning. In *AAAI*.

Wang, W.; Shen, J.; Lu, X.; Hoi, S. C. H.; and Ling, H. 2021a. Paying attention to video object pattern understanding. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(7): 2413–2428.

Wang, W.; Shen, J.; Porikli, F.; and Yang, R. 2018. Semi-supervised video object segmentation with super-trajectories. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(4): 985–998.

Wang, W.; Zhou, T.; Porikli, F.; Crandall, D.; and Van Gool, L. 2022. A survey on deep learning technique for video segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*

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

Wang, Z.; Zhao, H.; Li, Y.-L.; Wang, S.; Torr, P.; and Bertinetto, L. 2021b. Do Different Tracking Tasks Require Different Appearance Models? In *NeurIPS*.

Wei, S.-E.; Ramakrishna, V.; Kanade, T.; and Sheikh, Y. 2016. Convolutional pose machines. In *CVPR*.

Wiskott, L.; and Sejnowski, T. J. 2002. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4): 715–770.

Wojke, N.; Bewley, A.; and Paulus, D. 2017. Simple online and realtime tracking with a deep association metric. In *ICIP*.

Xie, Y.; Wang, Z.; and Ji, S. 2020. Noise2same: Optimizing a self-supervised bound for image denoising. In *NeurIPS*.

Xu, J.; and Wang, X. 2021. Rethinking self-supervised correspondence learning: A video frame-level similarity perspective. In *ICCV*.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How powerful are graph neural networks? In *ICLR*.

Xu, N.; Yang, L.; Fan, Y.; Yang, J.; Yue, D.; Liang, Y.; Price, B.; Cohen, S.; and Huang, T. 2018. Youtube-vos: Sequence-to-sequence video object segmentation. In *ECCV*.

Yang, L.; Fan, Y.; and Xu, N. 2019. Video Instance Segmentation. In *ICCV*.

Yin, Z.; and Shi, J. 2018. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*.

You, J.; Ying, R.; and Leskovec, J. 2019. Position-aware graph neural networks. In *ICML*.

Zeng, X.; Liao, R.; Gu, L.; Xiong, Y.; Fidler, S.; and Urtasun, R. 2019. Dmm-net: Differentiable mask-matching network for video object segmentation. In *CVPR*.

Zhao, Z.; Jin, Y.; and Heng, P.-A. 2021. Modelling Neighbor Relation in Joint Space-Time Graph for Video Correspondence Learning. In *ICCV*.

Zhou, Q.; Liang, X.; Gong, K.; and Lin, L. 2018. Adaptive Temporal Encoding Network for Video Instance-level Human Parsing. In *ACM MM*.