Deep Transport Network for Unsupervised Video Object Segmentation

Kaihua Zhang[1], Zicheng Zhao[2], Dong Liu[3], Qingshan Liu[1], Bo Liu[4*]
[1]ERCDF, Ministry of Education and School of Computing and Software, [2]School of Automation
Nanjing University of Information Science and Technology, Nanjing, China
[3]Netflix Inc. Los Gatos, CA, 95032, USA
[4]JD Finance America Corporation, Mountain View, CA, USA
{zhkhua,kfliubo}@gmail.com

Abstract

The popular unsupervised video object segmentation methods fuse the RGB frame and optical flow via a two-stream network. However, they cannot handle the distracting noises in each input modality, which may vastly deteriorate the model performance. We propose to establish the correspondence between the input modalities while suppressing the distracting signals via optimal structural matching. Given a video frame, we extract the dense local features from the RGB image and optical flow, and treat them as two complex structured representations. The Wasserstein distance is then employed to compute the global optimal flows to transport the features in one modality to the other, where the magnitude of each flow measures the extent of the alignment between two local features. To plug the structural matching into a two-stream network for end-to-end training, we factorize the input cost matrix into small spatial blocks and design a differentiable long-short Sinkhorn module consisting of a long-distant Sinkhorn layer and a short-distant Sinkhorn layer. We integrate the module into a dedicated two-stream network and dub our model TransportNet. Our experiments show that aligning motion-appearance yields the state-of-the-art results on the popular video object segmentation datasets.

1. Introduction

Video Object Segmentation (VOS) aims to track the moving objects with an accurate segmentation mask. It can be divided into two scenarios depending on whether the target objects are indicated at the test time. One scenario is the *Semi-supervised* VOS (SVOS) [59], where a model is trained over a training set, and at the test time the model is provided with the ground-truth mask on the first frame as prior to track the objects to be segmented in the subsequent frames. The other is called *Unsupervised* VOS (UVOS) or *primary object segmentation* [21][1], where no ground-truth mask is provided at the test time and there is no prior information about the target object. UVOS discovers the most salient, or primary, objects that move against a video's background[2], and all objects that consistently appear throughout the video with predominant motion is defined as one object [36, 2] (e.g., the person and the motorbike in Figure 1 are defined as one object). This is essentially different from the recently proposed task of *Unsupervised Multi-object Segmentation* [2], a variant of the conventional UVOS for segmenting separate objects in DAVIS-17 dataset [39], or the task of *Video Instance Segmentation* [60].

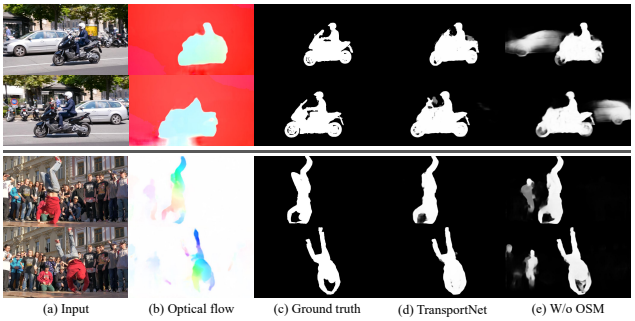

Figure 1. Examples of distracting signals in the appearance and motion input in UVOS. Our proposed TransportNet can generate accurate segmentation masks (column (d)) comparing to the method without Optimal Structural Matching (OSM) (column (e)).

We focus on the UVOS as it requires no user interactions. Since the target objects are unknown, the state-of-the-art UVOS methods rely on the motion cues (i.e.,

*Corresponding author. This work is supported in part by National Key Research and Development Program of China under Grant No. 2018AAA0100400, in part by the NSFC (61876088, 61825601), in part by the 333 High-level Talents Cultivation Project of Jiangsu Province (BRA2020291).

[1] Besides these two names, it is also called "zero-shot VOS" in the literature.
[2] In UVOS, the moving objects that are more likely to be followed by human gaze are referred to as "foreground" while the remaining regions (e.g., "people in crowds" or "still cars in the background") are referred to as "background" and not annotated as target objects in the ground truth [57].

optical flow) to find the primary objects to be segmented [51, 50, 19, 28, 27, 12]. A commonly-used architecture is a two-stream CNN, consisting of an appearance branch and a motion branch, which respectively takes the RGB frame and optical flow as parallel inputs [19]. To establish the deep interactions between appearance and motion, various network variants are adapted from the two-stream CNN, fusing the motion and appearance signals via sophisticated cross-modality learning module [70, 51, 50].

Despite promising performance, the existing methods are not able to well handle the *distracting signals* which may significantly deteriorate the model performance. In U-VOS, the distracting signals may originate from the RGB frame and/or from the optical flow. To illustrate the former, the top two rows in Figure 1 shows a video frame with the *motorbike* as the only primary target foreground object to be segmented on this dataset. As seen, the *car* appearing as a static background object is a distracting object which would bring ambiguity to the VOS model. On the other hand, as illustrated in the bottom two rows in Figure 1, the distracting signals in the optical flow are typically caused by the inaccurate flow estimation generated by models trained on synthetic videos [9]. When applying such models in real videos, the domain gap can cause the flow fields to contain significant noises, especially when the foreground object is nearly static. Such noises can be further amplified when there are unexpected camera and/or background object movements in the video. Therefore, it is not reliable to blindly fuse the appearance and motion features, and there is a demand to establish the correspondence between the flow vectors and the object instances while suppressing the distracting signals in the input.

A natural solution to align the motion and appearance is to compare their local features. The challenges lie in that we have no supervision on local motion-appearance correspondence for training and not all local features in one modality can find their counterparts in the other. We formulate the motion-appearance alignment as an instance of *Optimal Structure Matching* (OSM), and aim to discover the discriminative cross-modality patterns while minimizing the distracting noises in the input via structure learning. Given a video frame, we extract the dense local features from the optical flow and the RGB image, and obtain two complex structured representations, each consisting of a set of local building features. The *Wasserstein distance* [37] is then employed to compute the structural similarity between the two structured representations. The Wasserstein distance has the form of the *optimal transport* problem [37] which can find the global least-expensive flows to transport the local features in one modality to the local features in the other, leading to the minimum structural distance. The magnitude of each flow characterizes the degree of alignment between two local features and can be used to establish the motion-appearance correspondence. Since the matching process is reconstructing one structure against the other, the distracting noises not compatible with the holistic structural matching would end up with low magnitudes in their matching flows and be naturally filtered out.

To integrate the OSM into a two-stream CNN for end-to-end training, we design a differentiable neural network layer based on the *Sinkhorn* method [8], a solver used to optimize the Wasserstein distance. We notice that the Sinkhorn method involves computation/memory intensive matrix operations, hindering its applicability in VOS which typically requires high-resolution inputs and the stacking of multiple layers to ensure the performance. To this end, we propose a *Factorized Sinkhorn* method which factorizes a large input cost matrix into a number of small spatial blocks and performs structural matching within the long-distant and short-distant local blocks. The *long-distant matching* can well preserve the global structure information of the original complex representations while the *short-distant matching* can focus on the fine-grained details. By doing so, we not only speed up the optimization by $27.5\times$ but also improve the performance of UVOS by 3.7% in terms of mean $\mathcal{J}$ on FBMS dataset [33]. The two matching operations are implemented as two differentiable network layers termed as *Long-distant SinkHorn* (LSH) and *Short-distant SinkHorn* (SSH), which can be applied sequentially as a building block in a two-stream CNN for motion-appearance alignment. We plug our *Long-Short SinkHorh* (LSSH) block into a network architecture designed for UVOS and dub our network *TransportNet* to emphasize its origin from the optimal transport problem. We conduct extensive experiments on three popular benchmark datasets, demonstrating that our TransportNet yields the state-of-the-art performance.

Our contributions include: (1) a novel model exploiting the motion-appearance alignment for noise-tolerate U-VOS, (2) a unique OSM mechanism establishing the structural correspondence between motion and appearance signals while suppressing the noises, (3) a novel LSSH block to enable the structural matching in end-to-end training.

2. Related Work

Semi-supervised VOS. Numerous efforts are dedicated to model the object appearance for SVOS [34, 10, 45, 67, 18, 62]. To capture the evolution of the object mask, some recent works leverage RNN style network such as ConvLSTM and ConvGRU to model the long and/or short-term dynamics in video [52, 59, 51]. The other paradigm is to propagate the intermediate mask predictions on the previous frames to the current frame via memory network [34, 45, 31] or temporal bilateral network [20]. Besides modeling temporal dynamics, another line of research is to utilize the motion as complementary information and tackle the task via a two-stream network [4, 16, 58]. In Segflow [4], features

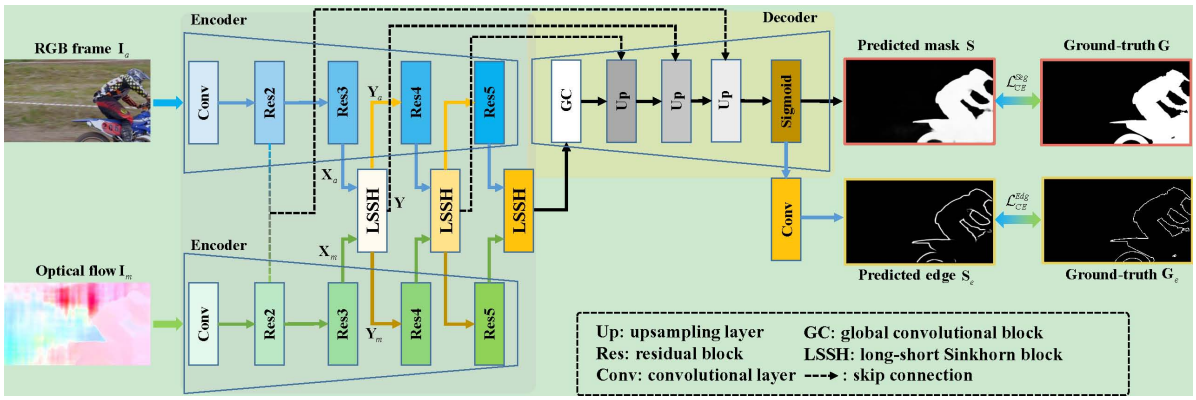

Figure 2. Network architecture of the proposed TransportNet for UVOS. The input RGB frame $\mathbf{I}_a$ and optimal flow $\mathbf{I}_m$ are passed through a two-stream ResNet backbone [15], extracting the appearance-motion features $\mathbf{X}_a$ and $\mathbf{X}_m$ at each residual stage (Res2∼Res5). The features derived from Res3 to Res5 are fed into the LSSHs to perform optimal structural matching, yielding the corresponding enhanced features $\mathbf{Y}_a$ and $\mathbf{Y}_m$. Then $\mathbf{Y}_a$ and $\mathbf{Y}_m$ are concatenated to yield the spatio-temporal feature representations $\mathbf{Y} = [\mathbf{Y}_a, \mathbf{Y}_m]$, which are further fed into the decoder via skip connections to produce the predicted object mask $\mathbf{S}$ and boundary map $\mathbf{S}_e$.

of object segmentation and optical flow are concatenated at different scales from mutual boosting. MoNet [58] was proposed to exploit the motion cue from optical flow to reinforce the representation of the target frame by integrating representations from its temporal neighbors. In contrast, we focus on UVOS without any prior. Instead of treating optical flows as reliable input [4, 58], we dynamically adjust the confidence of the flows by matching them to the appearance features via structure learning.

Unsupervised VOS. The popular UVOS methods leverage the appearance features of the video frames, and model their high-order relations [54, 31], dense pixel-wise correspondence [61, 27], or discriminative feature patterns [68]. Despite promising performance, they discard the motion signals which have been proved effective in the classic video analysis tasks [47, 69, 30]. A few recent methods [19, 28, 27, 12] suggest adopting the motion cue as additional information for inferring the object mask. LMP [50] trains a CNN, taking optical flow as input to separate the moving and non-moving regions, and then combines the results with objectness cues from SharpMask [38] to generate moving object mask. LVO [51] trains a two-stream fusion network, feeding the appearance features and optical flow features into a ConvGRU module [46] to generate the object mask. The recent effort in [70] points out that the motion cues in the existing methods are not adequately leveraged in that they are simply used either as extra input or as complementary features and therefore fail to capture the deep interactions between the two modalities. MATNet [70] was proposed to transform the appearance features via a motion-attentive transition block and generates a motion-attentive feature representations at each convolutional stage of the network. The drawback is that it blindly infers the region of interests by treating all the motion-appearance correspondences as equally reliable, which is different from our method trying to differentiate the reliability of the motion-appearance correspondences based on the matching scores.

Optimal Structural Matching. Various vision applications are formulated into an instance of structural matching via a solving an optimal transport problem. DeepEMD [67] was proposed to employ the Earth Mover's Distance [43] as a metric in supervised deep metric learning to perform structural matching between local patches of two images. In [44], the network flows between two complex structures are optimized to solve the multi-object tracking problem. SeLa [1] unifies clustering and representation learning on still images by extending the standard cross-entropy minimization to an optimal transport problem, and solves it by a variant of the Sinkhorn-Knopp algorithm [8]. These methods cannot be directly applied to our task since they are not tailored to match the fine-grained local features across modalities which demands a light and efficient optimizer facilitating the stacking of multiple OSM operations.

Interleaving. The factorization mechanism in our Factorized Sinkhorn method bears assembly with the general interleaving mechanism in network network architecture design, such as Shuffle operation in ShuffleNet [66], and the Interleaved group convolution [65]. Similar idea has recently been extended to the Interleaved Sparse Self-Attention [17] and Sparse Transformer [5]. Our work is different as our interleaving operation is explicitly driven by a structural matching objective.

3. TransportNet for UVOS

3.1. Network Architecture

Figure 2 illustrates the network architecture of our TransportNet. Specifically, given an input RGB frame $\mathbf{I}_a \in \mathbb{R}^{w\times h\times 3}$ and its optical flow map $\mathbf{I}_m \in \mathbb{R}^{w\times h\times 3}$, the encoder extracts their intermediate features $\mathbf{X}_a = [\mathbf{x}_{a,1}^\top; \ldots; \mathbf{x}_{a,N}^\top]$, $\mathbf{X}_m = [\mathbf{x}_{m,1}^\top; \ldots; \mathbf{x}_{m,N}^\top] \in \mathbb{R}^{N\times C}$ at each

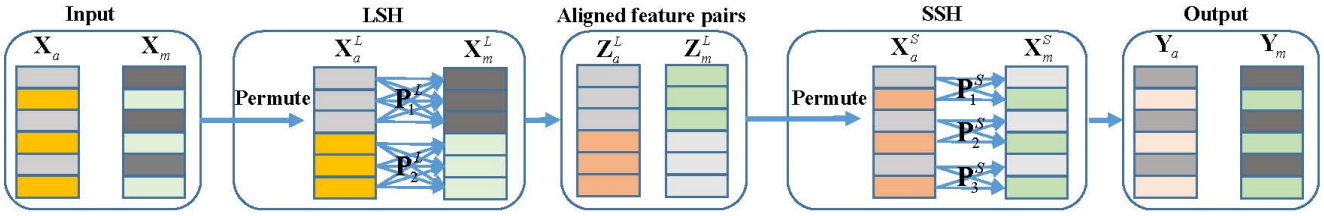

Figure 3. Pipeline of LSSH. The LSSH is composed of an LSH layer and an SSH layer. First, for each of the input $\{\mathbf{X}_a, \mathbf{X}_m\}$, we use the same color to represent the long-distant local features and $P = 2$ different colors to represent the short-distant ones, respectively. Then, the LSH layer first permutes $\{\mathbf{X}_a, \mathbf{X}_m\}$ to group features with the same color together, and then learns the optimal matching flows $\mathbf{P}_i^L, i = 1, \ldots, P$, for each group of features, outputting the aligned feature pairs $\{\mathbf{Z}_a^L, \mathbf{Z}_m^L\}$. The $\{\mathbf{Z}_a^L, \mathbf{Z}_m^L\}$ are then fed into the SSH layer, which are first permuted to generate $Q = 3$ groups of short-distant feature pairs with different colors together, and then learns the optimal matching flows $\mathbf{P}_i^S, i = 1, \ldots, Q$ for each group, outputting the finally aligned feature pairs $\{\mathbf{Y}_a, \mathbf{Y}_m\}$.

Residual block, where $\{\mathbf{x}_{a,i}, \mathbf{x}_{m,i} \in \mathbb{R}^C\}$ denotes the i-th appearance-motion local feature pair, $N = WH$ denotes the number of features, W and H are the width and height of the feature map, and C denotes the number of feature channels. Afterwards, $\mathbf{X}_a, \mathbf{X}_m$ are fed into an LSSH module $\mathcal{N}_{\text{LSSH}}$ (see Figure 3), to produce the aligned appearance and motion features

$$\{\mathbf{Y}_a, \mathbf{Y}_m\} = \mathcal{N}_{\text{LSSH}}(\mathbf{X}_a, \mathbf{X}_m). \tag{1}$$

Inside the LSSH module, we employ the Wasserstein distance [14] to measure the structural similarity between $\mathbf{X}_a$ and $\mathbf{X}_m$. To efficiently optimize the Wasserstein distance, we design a Factorized Sinkhorn composed of an LSH layer $\mathcal{N}_{\text{LSH}}$, followed by an SSH layer $\mathcal{N}_{\text{SSH}}$. We concatenate $\mathbf{Y}_a, \mathbf{Y}_m$ to produce the enhanced spatio-temporal representation $\mathbf{Y} = [\mathbf{Y}_a, \mathbf{Y}_m] \in \mathbb{R}^{N \times 2C}$, and feed $\mathbf{Y}$ into the corresponding feature pyramid module at the decoder layer with skip connections. Finally, the decoder layer produces the predicted segmentation map $\mathbf{S} \in \mathbb{R}^{w \times h}$.

3.2. Optimal Structural Matching

3.2.1 Wasserstein Distance

Given the feature point sets $\{\mathbf{X}_a, \mathbf{X}_m\}$, the squared Wassertein distance is defined as [14]

$$W_2^2(\mathbf{X}_a, \mathbf{X}_m) = \min_{\mathbf{P} \in \mathcal{P}_N} \sum_{i,j} \mathbf{P}(i,j)\mathbf{C}(i,j), \tag{2}$$

where $\mathbf{P}$ is a transition matrix with each element $\mathbf{P}(i,j)$ representing the matching flow of the appearance-motion feature pair $\{\mathbf{x}_{a,i}, \mathbf{x}_{m,j}\}$. The set $\mathcal{P}_N = \{\mathbf{P} \in (\mathbb{R}^+)^{N \times N}, \mathbf{P}\mathbf{1}_N = \mathbf{1}_N, \mathbf{P}^\top \mathbf{1}_N = \mathbf{1}_N\}$, where $\mathbf{1}_N$ is an N-dimensional all-one vector. $\mathbf{C}$ is the cost matrix with $\mathbf{C}(i,j) = \|\mathbf{x}_{a,i} - \mathbf{x}_{m,j}\|_2^2$, which is the L_2 distance between two feature points. Since we use the normalized features, $\mathbf{C}(i,j)$ can be reformulated as

$$\mathbf{C}(i,j) = 2 - 2\mathbf{x}_{a,i}^\top \mathbf{x}_{m,i}. \tag{3}$$

From the optimal transport theory [25, 14], the W_2^2 in (2) is the minimum cost induced by the optimal transport plan, and the $\mathbf{C}(i,j)$ denotes the cost to move a probability mass from $\mathbf{x}_{a,i}$ to $\mathbf{x}_{m,j}$, i.e., $p_{a,i} \to p_{m,j}$, satisfying

$$\sum_{i=1}^{N} p_{a,i}\delta(\mathbf{x} - \mathbf{x}_{a,i}) = 1, \sum_{i=1}^{N} p_{m,i}\delta(\mathbf{x} - \mathbf{x}_{m,i}) = 1, \tag{4}$$

where $\delta(\cdot)$ is the Dirac function that satisfies $\delta(\mathbf{x}) = 1$ if $\mathbf{x} = \mathbf{0}$, and otherwise $\delta(\mathbf{x}) = 0$. Note that if we assume that the unary matchabilities are uniformly distributed [3], that is $\forall i, j$, $p_{a,i} = p_{m,j} = \frac{1}{N}$, meaning that each feature point has the same prior likelihood of matching, then the global optimal matching flows $\mathbf{P}$ in (2) is a permutation matrix. In this case, the optimal transport plan is equal to solving an optimal assignment problem that is a linear program [6].

Since problem (2) is a linear program, it can be readily solved with combinational algorithms including simplex methods and their variants such as Hugarian or relaxation algorithms [37, 64]. However, it has been shown that the best computational complexity of these methods is $O(2N^3 \log 2N)$ [37]. This hinders the method from handling large-scale datasets. Recently, Cuturi [8] presents an efficient Sinkhorn method to solve the transport plan, which has several orders of magnitude faster than the former transport solvers, thereby attracting much attention in a variety of vision tasks [3, 6, 7, 53, 63] for optimal structural matching. To further efficiently optimize the problem (2), we design a *Factorized Sinkhorn* method that includes an LSH layer followed by an SSH layer. In the following, we first present the Sinkhorn layer, and then introduce the LSH and SSH layers.

3.2.2 SinkHorn (SH) Layer

With the cost matrix $\mathbf{C}$ in (3) as input, we design an SH layer that leverages the Sinkhorn method [8] to model the structural feature matching as a linear assignment problem. The SH layer relaxes the discrete assignment constraint as a doubly-stochastic matrix, which can be seen as a continuous relaxation of the permutation matrix $\mathbf{P}$ that globally optimizes the W_2^2 in (2). The Sinkhorn alternatively takes row- and column-normalization until convergence. Given

Figure 4. Examples of the matching results by LSH (blue arrow) and SSH (orange arrows), respectively. The starting points of arrows denote the selected locations on the RGB frames, and the ending points represent the top-k matching locations (i.e., those with the largest matching scores) on the optical flow maps estimated by LSH ($k = 1$) and SSH ($k = 2$), respectively.

$\mathbf{C}_{(0)} = \mathbf{C}$, the k-th iteration of the Sinkhorn operator is

$$\begin{aligned} \widetilde{\mathbf{C}}_{(k)} &= \mathbf{C}_{(k-1)} \oslash (\mathbf{C}_{(k-1)} \mathbf{1}_N \mathbf{1}_N^\top), \\ \mathbf{C}_{(k)} &= \widetilde{\mathbf{C}}_{(k)} \oslash (\mathbf{1}_N \mathbf{1}_N^\top \widetilde{\mathbf{C}}_{(k)}), \end{aligned} \tag{5}$$

where $\oslash$ denotes the element-wise division. After convergence, we obtain a doubly-stochastic matrix $\mathbf{P}$ that is our optimal matching flows. We modularize the SH layer as

$$\mathbf{P} = \mathcal{N}_{\mathrm{SH}}(\mathbf{C}). \tag{6}$$

The SH layer is differentiable since its operation in (5) only contains matrix-vector multiplication and element-wise division, which can be readily plugged into the vanilla deep neural networks for end-to-end training. The backward gradient of the SH layer can be readily derived from [53], which can be readily implemented with automatic differentiation in PyTorch [35].

The dense cost matrix $\mathbf{C}$ introduces heavy computation/memory cost when optimizing the SH layer (6), preventing using high-resolution inputs and stacking multiple SH layers that are essential for high performance in various vision tasks. To reduce computation/memory cost, existing works [63, 53, 7] for feature matching only plug in one SH layer at the end of their network architectures. We address this issue by factorizing the $\mathbf{C}$ into two sparse block distance matrices $\mathbf{C}^L$ and $\mathbf{C}^S$, which capture the long- and short-range dependencies between spatial locations on the feature map, respectively. We then perform structural matching within the long-distant and short-distant local blocks. This matrix factorization method significantly reduces the computation/memory cost, making our network be able to stack 3 SH layers with high-resolution inputs.

3.2.3 Long-distant SinkHorn (LSH) Layer

The long-distant cost matrix $\mathbf{C}^L \in \mathbb{R}^{N \times N}$ is designed to capture the interactions between any pair-wise appearance and motion features at long-distant spatial locations on the feature map. As shown in Figure 3, to group together the features at long-distant locations, we first permute the appearance feature map $\mathbf{X}_a \in \mathbb{R}^{N \times C}$ to generate $\mathbf{X}_a^L = \mathrm{permute}(\mathbf{X}_a)$. Then, we equally divide the $\mathbf{X}_a^L$ into P parts, and each part has Q feature vectors. Then, the $\mathbf{X}_a^L$ can be rewritten as $\mathbf{X}_a^L = [\mathbf{X}_{a,1}^L; \mathbf{X}_{a,2}^L; \ldots; \mathbf{X}_{a,P}^L]$, where each $\mathbf{X}_{a,p}^L \in \mathbb{R}^{Q \times C}$ is from Q long-distant spatial locations. Similar operations are done for the motion feature map, yielding the corresponding motion features $\mathbf{X}_m^L = [\mathbf{X}_{m,1}^L; \mathbf{X}_{m,2}^L; \ldots; \mathbf{X}_{m,P}^L]$, where each $\mathbf{X}_{m,p}^L \in \mathbb{R}^{Q \times C}$. Then, we define the $\mathbf{C}^L$ as a sparse block matrix

$$\mathbf{C}^L = \mathrm{diag}\big(\mathbf{C}_1^L, \mathbf{C}_2^L, \ldots, \mathbf{C}_P^L\big), \tag{7}$$

where each $\mathbf{C}_p^L = \mathbf{X}_{a,p}^L \mathbf{X}_{m,p}^{L^\top} \in \mathbb{R}^{Q \times Q}$ is a small cost matrix based on all spatial locations from $\mathbf{X}_{a,p}^L$ and $\mathbf{X}_{m,p}^L$, and diag denotes a diagonal block matrix operator. We apply the SH layer (6) on each $\mathbf{C}_p^L$, and obtain the optimal matching flows $\mathbf{P}^L = \mathrm{diag}\big(\mathbf{P}_1^L, \mathbf{P}_2^L, \ldots, \mathbf{P}_P^L\big)$. Finally, the aligned appearance and motion features can be computed as

$$\mathbf{Z}_a^L = \mathbf{P}^L \mathbf{X}_a^L, \mathbf{Z}_m^L = \mathbf{P}^L \mathbf{X}_m^L. \tag{8}$$

3.2.4 Short-distant SinkHorn (SSH) Layer

As aforementioned, the short-distant cost matrix $\mathbf{C}^S$ captures the interactions between the appearance-motion features at spatial locations that have short spatial distances.

As shown in Figure 3, we leverage another permutation on the output feature maps $\mathbf{Z}_a^L$ and $\mathbf{Z}_m^L$ from the LSH layer, yielding $\mathbf{X}_a^S$ and $\mathbf{X}_m^S$. Then, we equally divide the $\mathbf{X}_a^S$ into Q parts, and each part has P neighboring feature vectors. The $\mathbf{X}_a^S$ can be rewritten as $\mathbf{X}_a^S = [\mathbf{X}_{a,1}^S; \mathbf{X}_{a,2}^S; \ldots; \mathbf{X}_{a,Q}^S]$, where each $\mathbf{X}_{a,q}^S \in \mathbb{R}^{P \times C}$ is from P short-distant locations. Similar operations are performed for the motion feature map, yielding the corresponding motion features $\mathbf{X}_m^S = [\mathbf{X}_{m,1}^S; \mathbf{X}_{m,2}^S; \ldots; \mathbf{X}_{m,Q}^S]$, where each $\mathbf{X}_{m,q}^S \in \mathbb{R}^{P \times C}$. Then, similar to the long-distant cost matrix $\mathbf{C}^L$, we define the $\mathbf{C}^S$ as a sparse block matrix

$$\mathbf{C}^S = \mathrm{diag}\big(\mathbf{C}_1^S, \mathbf{C}_2^S, \ldots, \mathbf{C}_Q^S\big), \tag{9}$$

where each $\mathbf{C}_q^S = \mathbf{X}_{a,q}^S \mathbf{X}_{m,q}^{S^\top} \in \mathbb{R}^{P \times P}$. We apply the SH layer (6) on $\mathbf{C}^S$, yielding the optimal matching flows $\mathbf{P}^S = \mathrm{diag}\big(\mathbf{P}_1^S, \mathbf{P}_2^S, \ldots, \mathbf{P}_Q^S\big)$. Finally, we calculate the aligned appearance and motion feature maps as

$$\mathbf{Y}_a = \mathbf{P}^S \mathbf{X}_a^S, \mathbf{Y}_m = \mathbf{P}^S \mathbf{X}_m^S. \tag{10}$$

Figure 4 visualizes two matching examples using LSH and SSH, where the left example selects one point from the noisy background while the right one selects one point from the foreground target. The top-1 and top-2 matching locations with the largest matching flows in $\mathbf{P}^L$ and $\mathbf{P}^S$ are shown as the ending points of arrows, where we can observe that both LSH and SSH can establish the accurate correspondences with large probabilities.

Theoretical Justification of LSSH. Theoretically, the merit of LSSH is inspired by the classic convolution operator proposed in LeNet [24]. As in Figure 3, LSSH performs sparse matching between appearance and motion features and the rationality lies in that the spatial correlation between them is local (liken to applying a small kernel in conv operator to model local spatial correlation). Contrarily, SH does dense matching that not only ignores the local correlation but also easily causes overfitting (liken to the fully-connected layer).

3.3. Decoder Network

The decoder network is similar to U-Net [42], which uses skip-connections to fuse the multi-scale spatio-temporal features from the encoder to the decoder. The fused feature maps are gradually upscaled by a factor of two at a time, and they are then concatenated with the feature maps of the next layer. Finally, the aggregated features are fed into a convolutional layer followed by a Softmax layer and a convolutional layer to predict the object mask $\mathbf{S}$ and object edge mask $\mathbf{S}_e$ respectively. The loss function for the network optimization is defined as the cross-entropy loss that aims to pixel-wisely classifying the objects and their boundaries

$$\mathcal{L} = \mathcal{L}_{CE}^{Seg} + \lambda \mathcal{L}_{CE}^{Edg}, \tag{11}$$

where $\mathcal{L}_{CE}^{Seg} = -\sum_{ij} \mathbf{G}(i,j) \log \mathbf{S}(i,j)$ and $\mathcal{L}_{CE}^{Edg} = -\sum_{ij} \mathbf{G}_e(i,j) \log \mathbf{S}_e(i,j)$, where $\mathbf{G} \in \{0,1\}^{w \times h}$ denotes the ground-truth mask and $\mathbf{G}_e$ denotes its corresponding edge mask that is easily estimated by the Sobel operator. λ is a tradeoff parameter empirically setting to 1.0 according to the validation performance.

4. Experiments

4.1. Implementation Details

The backbones of the appearance and motion streams in our TransportNet are ResNet101 and ResNet50 [15] respectively. The LSSH is plugged at the 3-rd, 4-th, and 5-th residual blocks of the ResNet models. In each LSSH, the feature map with size $W \times H \times C$ is set to $P = 8$ partitions according to validation performance, and each contains $Q = WH/P$ features.

Our experiments follow the common practices as in [70]. The training set consists of two parts: a) all the training data in the DAVIS-16 [36], which includes 30 videos with $2,000$ frames; b) a subset of $8,000$ frames selected from the training set of Youtube-VOS [59], which is obtained by sampling one frame every 10 frames in each video. We use $10,000$ frames for training, which are much less than the recent MATNet [70] that uses $14,000$ training frames. All images are unified to $512 \times 512 \times 3$, and the PWC-Net [49] is adopted to estimate their optical flows due to its high efficiency and accuracy. Note that using PWCNet and ResNet as the network backbone is a common practice in UVOS [70, 31, 61], and our work follows this practice to ensure fair comparison. The network is trained with the Adam optimizer [23] with an initial learning rate of 1e-4 for the encoder, and 1e-3 for the decoder. We set the batch size, momentum and weight decay to 2, 0.9, and 1e-5 respectively, and augment the training samplings with horizontal flip and rotations that cover a range of degrees $(-10, 10)$.

After training, we test the model on unseen videos for evaluation. We resize each frame to 512×512, and feed it and its corresponding optical flow map into the trained model to generate the segmentation map, which is then binarized by a threshold $= 0.5$ to directly produce the binary segmentation mask without any further post-processing.

The TransportNet is implemented in PyTorch [35] on an Nvidia GTX 2080Ti GPU. For each test image of size $512 \times 512 \times 3$, the forward inference of our TransportNet takes $\sim 0.08s$, while the optical flow estimation by PWC-Net takes $\sim 0.2s$ (offline independent of inference). Even adding the offline optical flow estimation time, our model takes about $0.08s + 0.2s = 0.28s$, which is still competitive to the online inference time of DFNet [68] ($0.28s$/image).

4.2. Datasets and Evaluation Metrics

We conduct experiments on four popular benchmark datasets including DAVIS-16 [36], FBMS [33], ViSal [55] and Youtube-Objects [40] (due to space limitation, the results on Youtube-Objects are put in the supplemental material, in which our method achieves the SOTA performance). Note that most of the existing UVOS works use these datasets as the testbed [61, 68, 32] and we follow this practice to ensure fair comparison. Though there are other VOS datasets such as Youtube-VOS [59] and DAVIS-17 [39], they are either only used for evaluating semi-supervised VOS [34, 62] (the former), or better fitting for a recently proposed new task of multi-object segmentation [2] (the latter, because it provides instance annotations). We thus choose not to use them in our experiments.

DAVIS-16 totally consists of 50 videos, including 30 videos for training and 20 for testing. Each frame offers a pixel-wisely annotated mask for foreground objects. In this dataset, we leverage three evaluation metrics provided by [36], including a) region similarity $\mathcal{J}$, b) boundary accuracy $\mathcal{F}$, c) overall $\mathcal{J}\&\mathcal{F}$ score that is the average of $\mathcal{J}$ and $\mathcal{F}$ scores. Besides, we also report the salient object detection results on this dataset in terms of the Mean Absolution Error (MAE) and the F-measure $\mathcal{F}_m$.

FBMS consists of 59 video sequences with 29 training videos and 30 test ones. The ground-truth annotations for every 20 frame are provided, producing a total of 720 annotated frames in the entire dataset. We evaluate on the test set, and the main evaluation metrics are region similarity $\mathcal{J}$,

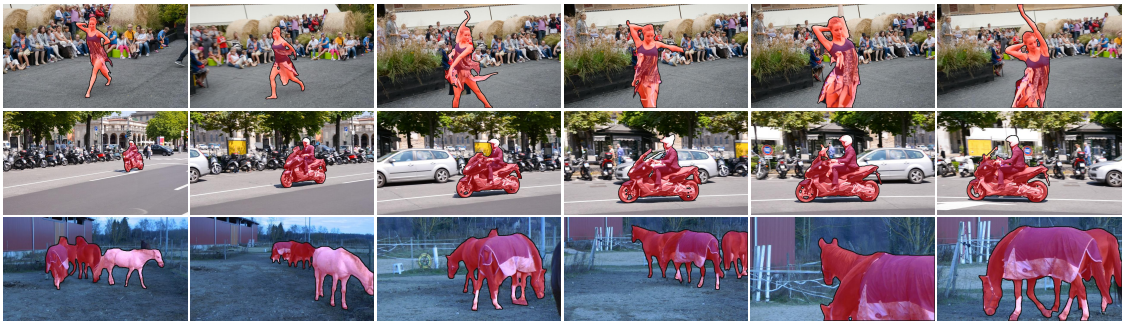

Figure 5. Qualitative results. From top to bottom: *dance-twirl* from DAVIS-16, *scooter-black* from DAVIS-16, and *horse04* from FBMS.

MAE and F-measure $\mathcal{F}_m$.

ViSal is designed for video saliency task that contains a collection of 17 videos with a diverse set of objects and backgrounds, varying in length from 30 to 100 frames. It has 193 manually-annotated frames. The whole dataset is used for evaluation in terms of MAE and F-measure $\mathcal{F}_m$.

4.3. Comparison with the State-of-the-arts

Quantitative Results. Table 1 lists the quantitative comparison results of our TransportNet against the state-of-the-art UVOS methods on DAVIS-16 and FBMS. Note that it is unfair to compare our UVOS results to the results achieved by semi-supervised VOS methods such as STM [34]. Our TransportNet achieves the best $\mathcal{J}\&\mathcal{F} = 84.8\%$ over the existing state-of-the-arts. Moreover, it also achieves the highest performance in terms of all other evaluation metrics on both datasets. Moreover, the TransportNet reaches a new state-of-the-art result on the test set of FBMS with $\mathcal{J} = 78.7\%$, a significant gain of 2.6% over the second best performing method MATNet with $\mathcal{J} = 76.1\%$. This demonstrates that the TransportNet can produce high quality segmentation masks by aligning the appearance and motion features through optimal structural matching. On the contrary, a variety of competing methods (i.e., MATNet, COSNet, and AnDiff) apply post-processing techniques such as CRFs or instance pruning to improve the performance, which introduce more computational cost. Without this post-processing step, 3DC-Seg [32] is the best-performing existing method that leverages 3D convolutions for UVOS, outperforming those based on 2D convolutions by a large margin. 3DC-Seg uses a backbone of 3D ResNet-152 pretrained on IG-65M [13] and Kinetics [22], and then trains the model using COCO Instance Segmentation dataset [29], YouTube-VOS [59], and DAVIS-16 [36]. Notwithstanding, the proposed TransportNet outperforms 3DC-Seg in terms of all evaluation metrics with fewer training data and less computation overhead due to the use of 2D convolutions. This also verifies the effectiveness of the proposed OSM mechanism in our TransportNet that helps learn a strong spatial-temporal representation that is essential to produce high-quality segmentation masks.

Table 2 shows the qualitative results of the TransportNet compared to the state-of-the-arts for the task of video saliency on DAVIS16, FBMS, and ViSal datasets. The task of video saliency is similar to UVOS, where the definition of salient objects in ViSal [55] is related to the foreground objects in DAVIS-16 and FBMS. As shown in Table 2, the proposed TransportNet achieves new state-of-the-art on DAVIS-16 and ViSal datasets in terms of all evaluation metrics, especially for the $\mathcal{F}_m$ scores. Our TransportNet achieves $\mathcal{F}_m = 88.5\%$ and $\mathcal{F}_m = 95.3\%$ on FBMS and ViSal respectively, with a significant gain of 4% and 3.1% over the competing counterpart 3DC-Seg with $\mathcal{F}_m = 84.5\%$ and 92.2%. Notably, despite only being trained for the UVOS task, our TransportNet outperforms the state-of-the-art video saliency method TENet [41] in terms of both MAE and $\mathcal{F}_m$ on DAVIS-16 and ViSal, and achieves very competitive results on FBMS (MAE $= 0.045$ vs. 0.026, $\mathcal{F}_m = 88.5\%$ vs. 89.7%). The **speed comparison** and **precision-recall curve plot** can be found in the supplemental material.

Method	DAVIS16 $\mathcal{J}\&\mathcal{F}\uparrow$	DAVIS16 $\mathcal{J}\uparrow$	DAVIS16 $\mathcal{F}\uparrow$	FBMS $\mathcal{J}\uparrow$
LMP(CVPR17) [50]	68.0	70.0	65.9	-
LVO(ICCV17) [51]	74.0	75.9	72.1	-
PDB(ECCV18) [48]	75.9	77.2	74.5	74.0
MBNM(ECCV18) [28]	79.5	80.4	78.5	73.9
AGS(CVPR19) [57]	78.6	79.7	77.4	-
COSNet(CVPR19) [31]	80.0	80.5	79.4	75.6
AGNN(ICCV19) [54]	79.9	80.7	79.1	-
AnDiff(ICCV19) [61]	81.1	81.7	80.5	-
EpO+(WACV20) [11]	78.1	80.6	75.5	-
MATNet(AAAI20) [70]	81.6	82.4	80.7	<u>76.1</u>
DFNet(ECCV20) [68]	82.6	83.4	81.8	-
3DC-Seg(BMVC20) [32]	<u>84.5</u>	<u>84.3</u>	<u>84.7</u>	-
TransportNet	**84.8**	**84.5**	**85.0**	**78.7**

Table 1. Quantitative results on the validation sets of DAVIS-16 and FBMS. For FBMS, we report $\mathcal{J}$ results. The best and second-best results are highlighted using **bold** and <u>underline</u>.

Qualitative Results. Figure 5 shows some qualitative results from DAVIS-16 and FBMS datasets. Specifically, *dance-twirl* and *scooter-black* videos are from DAVIS-16, where the foreground objects suffer from severe deformation, scale variation, and background clutter. *horse04* video is from FBMS, where it contains multiple horses as fore-

Method	DAVIS16		FBMS		ViSal	
	MAE↓	$\mathcal{F}_m$↑	MAE↓	$\mathcal{F}_m$↑	MAE↓	$\mathcal{F}_m$↑
FCNS*(TIP17) [56]	0.053	72.9	0.100	73.5	0.041	87.7
FGRNE*(CVPR18) [26]	0.043	78.6	0.083	77.9	0.040	85.0
TENet*(ECCV20) [41]	0.019	90.4	**0.026**	**89.7**	0.014	94.9
MBNM(ECCV18) [28]	0.031	86.2	0.047	81.6	0.047	-
PDB(ECCV18) [48]	0.030	84.9	0.069	81.5	0.022	91.7
AnDiff(ICCV19) [61]	0.044	80.8	0.064	81.2	0.030	90.4
DFNet(ECCV20) [68]	0.018	89.9	0.054	83.3	0.017	92.7
3DC-Seg(BMVC20) [32]	0.015	91.8	0.048	84.5	0.019	92.2
TransportNet	**0.013**	**92.8**	0.045	88.5	**0.012**	**95.3**

Table 2. Quantitative results in terms of MAE and maximum F-measure on DAVIS16, FBMS and ViSal datasets. The best and second-best results are highlighted using **bold** and underline. * means that the method is tailored to the video saliency task.

grounds to be segmented that undergo significant non-rigid deformations. We see TransportNet can well handle these challenges and delineate the targets with accurate contours. More visual examples are in the supplemental material.

4.4. Ablation Study

Table 3 lists the ablation results of our module variants on DAVIS-16 and FBMS in terms of mean $\mathcal{J}$, which are categorized into three groups to verify the effectiveness of LSH&SSH, LSSH location, and edge loss, respectively.

Module Variants							DAVIS-16	FBMS
SH	LSH	SSH	Res3	Res4	Res5	Edge loss	mean $\mathcal{J}$ ↑	mean $\mathcal{J}$ ↑
✓			✓	✓	✓	✓	83.4	75.0
	✓		✓	✓	✓	✓	82.4	73.2
		✓	✓	✓	✓	✓	83.9	77.4
						✓	80.4	68.2
	✓	✓			✓	✓	83.3	72.7
	✓	✓		✓	✓	✓	83.8	76.4
	✓	✓	✓	✓	✓		84.3	76.3
	✓	✓	✓	✓	✓	✓	**84.5**	**78.7**

Table 3. Ablations on DAVIS-16 and FBMS. 'Res3', 'Res4', and 'Res5' represent the locations in the network where LSSH is plugged.

Effect of LSH&SSH. We take the SH introduced in Sec 3.2.2 as a baseline. The model with SH achieves mean $\mathcal{J} = 83.4\%$ and 75.0% on DAVIS-16 and FBMS, which are 1% and 1.8% higher than our model with LSH. The reason is that the LSH fails to capture the local fine-grained details that are essential to perform accurate matching. On contrast, with SSH, our model achieves the mean $\mathcal{J} = 82.4\%$ and 73.2%, outperforming the baseline by 0.5% and 2.4% on DAVIS-16 and FBMS, respectively. This demonstrates the effectiveness of SSH that is able to perform more robust matching than SH. The SSH does robust matching by searching local regions, which can naturally avoid the long-range noisy interferences that will degrade the baseline model with SH. Finally, our model with LSSH achieves the best mean $\mathcal{J} = 84.5\%$ and 78.7%, with obvious gains of 1.1% and 3.7% over the baseline, demonstrating the effectiveness of the proposed factorized SH strategy to handle the distracting signals.

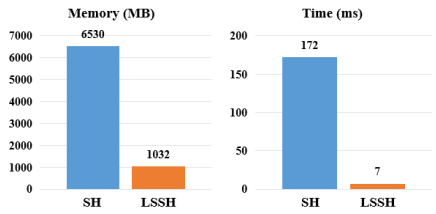

Figure 6. GPU memory/Time comparison between SH and LSSH.

Effect of LSSH location. Without plugging any LSSH, our model achieves mean $\mathcal{J} = 80.4\%$ and 68.2% on DAVIS-16 and FBMS, lower than only plugging one LSSH after Res5 by a large margin of 2.9% and 4.5%. We then further plug the LSSH after Res4, achieving the mean $\mathcal{J} = 83.8\%$ and 72.7%, with a gain of 0.5% and 3.7% against the former counterpart. By continuing to plug the LSSH after Res3, the performance is improved to mean $\mathcal{J} = 84.5\%$ and 78.7%. To better balance between the performance and computation/memory cost, our TransportNet contains three LSSHs plugged after Res3, Res4, and Res5, respectively.

Effect of Edge Loss. Without the edge loss, our model achieves mean $\mathcal{J} = 84.3\%$ and 76.3% on DAVIS-16 and FBMS, respectively. The performance is lower than our TransportNet with edge loss by 0.2% and 2.4%. This verifies the effectiveness of the boundary-enhanced information that can help to produce more accurate segmentation masks.

Efficiency Comparison. Figure 6 shows the results of memory and time cost between our proposed LSSH and the SH. The processing input feature map is of size $128 \times 128 \times 2048$. We can observe that our LSSH only uses 15.8% GPU memory while being nearly $24.5\times$ faster compared to the SH. This verifies the efficiency of our solver to the task of optimal structural matching.

5. Conclusions

In this paper, we presented a Wasserstein distance-based optimal structural matching network, termed as TransportNet for UVOS. The TransportNet has a two-stream structure that establishes the correspondence between the input modalities of RGB frame and optical flow while suppressing the distracting signals via optimal structural matching. To be more specific, the Wasserstein distance has been employed to compute the global optimal flows to transport the features in one modality to the other. To plug the structural matching into the network for efficient end-to-end training, we have factorized the input cost matrix into small spatial blocks and designed a differentiable LSSH module consisting of an LSH layer and an SSH layer. Extensive experiments on DAVIS-16, FBMS, and ViSal have demonstrated favorable performance of our TransportNet against the state-of-the-art methods in terms of all evaluation metrics.

References

[1] Yuki M. Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2020. 3

[2] Sergi Caelles, Jordi Pont-Tuset, Federico Perazzi, Alberto Montes, Kevis-Kokitsi Maninis, and Luc Van Gool. The 2019 davis challenge on vos: Unsupervised multi-object segmentation. *arXiv preprint arXiv:1905.00737*, 2019. 1, 6

[3] Dylan Campbell, Liu Liu, and Stephen Gould. Solving the blind perspective-n-point problem end-to-end with robust differentiable geometric optimization. *arXiv preprint arXiv:2007.14628*, 2020. 4

[4] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang. Segflow: Joint learning for video object segmentation and optical flow. In *ICCV*, 2017. 2, 3

[5] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. 3

[6] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *TPMAI*, 2016. 4

[7] R. Santa Cruz, B. Fernando, A. Cherian, and Stephen Gould. Visual permutation learning. *TPAMI*, 2019. 4, 5

[8] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*, 2013. 2, 3, 4

[9] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 2

[10] Kevin Duarte, Yogesh S. Rawat, and Mubarak Shah. Capsulevos: Semi-supervised video object segmentation using capsule routing. In *ICCV*, 2019. 2

[11] Muhammad Faisal, Ijaz Akhter, Mohsen Ali, and Richard Hartley. Epo-net: Exploiting geometric constraints on dense trajectories for motion saliency. In *WACV*, 2020. 7

[12] Aikaterini Fragkiadaki, Bryan Seybold, Cordelia Schmid, Rahul Sukthankar, Sudheendra Vijayanarasimhan, and Susanna Ricco. Self-supervised learning of structure and motion from video. In *arxiv (2017)*, 2017. 2, 3

[13] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *CVPR*, 2019. 7

[14] Edouard Grave, Armand Joulin, and Quentin Berthet. Unsupervised alignment of embeddings with wasserstein procrustes. In *AISTATS*, 2019. 4

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 6

[16] Ping Hu, Gang Wang, Xiangfei Kong, Jason Kuen, and Yap-Peng Tan. Motion-guided cascaded refinement network for video object segmentation. In *CVPR*, 2018. 2

[17] Lang Huang, Yuhui Yuan, Jianyuan Guo, Chao Zhang, Xilin Chen, and Jingdong Wang. Interlaced sparse self-attention for semantic segmentation. In *arXiv*, 2019. 3

[18] Xuhua Huang, Jiarui Xu, Yu-Wing Tai, and Chi-Keung Tang. Fast video object segmentation with temporal aggregation network and dynamic template matching. In *CVPR*, 2020. 2

[19] Suyog Jain, Bo Xiong, and Kristen Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. In *CVPR*, 2017. 2, 3

[20] Varun Jampani, Raghudeep Gadde, and Peter V. Gehler. Video propagation networks. In *CVPR*, 2017. 2

[21] Yeong Jun Koh and Chang-Su Kim. Primary object segmentation in videos based on region augmentation and reduction. In *CVPR*, 2017. 1

[22] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 7

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[24] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998. 6

[25] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *CVPR*, 2019. 4

[26] Guanbin Li, Yuan Xie, Tianhao Wei, Keze Wang, and Liang Lin. Flow guided recurrent neural encoder for video salient object detection. In *CVPR*, 2018. 8

[27] Siyang Li, Bryan Seybold, Alexey Vorobyov, Alireza Fathi, Qin Huang, and C.-C. Jay Kuo. Instance embedding transfer to unsupervised video object segmentation. In *CVPR*, 2018. 2, 3

[28] Siyang Li, Bryan Seybold, Alexey Vorobyov, Xuejing Lei, and C-C Jay Kuo. Unsupervised video object segmentation with motion-based bilateral networks. In *ECCV*, 2018. 2, 3, 7, 8

[29] 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 *ECCV*, 2014. 7

[30] Ding Liu, Zhaowen Wang, Yuchen Fan, Xianming Liu, Zhangyang Wang, Shiyu Chang, and Thomas Huang. Robust video super-resolution with learned temporal dynamics. In *ICCV*, 2017. 3

[31] Xiankai Lu, Wenguan Wang, Chao Ma, Jianbing Shen, Ling Shao, and Fatih Porikli. See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In *CVPR*, 2019. 2, 3, 6, 7

[32] Sabarinath Mahadevan, Ali Athar, Aljoša Ošep, Sebastian Hennen, Laura Leal-Taixé, and Bastian Leibe. Making a case for 3d convolutions for object segmentation in videos. In *BMVC*, 2020. 6, 7, 8

[33] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *TPAMI*, 2013. 2, 6

[34] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019. 2, 6, 7

[35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 5, 6

[36] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 1, 6, 7

[37] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *FTML*, 2019. 2, 4

[38] Pedro O Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. Learning to refine object segments. In *ECCV*, 2016. 3

[39] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. 2017. 1, 6

[40] Alessandro Prest, Christian Leistner, Javier Civera, Cordelia Schmid, and Vittorio Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012. 6

[41] Sucheng Ren, Chu Han, Xin Yang, Guoqiang Han, and Shengfeng He. Tenet: Triple excitation network for video salient object detection. In *ECCV*, 2020. 7, 8

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 6

[43] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *IJCV*, 2000. 3

[44] Samuel Schulter, Paul Vernaza, Wongun Choi, and Manmohan Chandraker. Deep network flow for multi-object tracking. In *CVPR*, 2017. 3

[45] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Kernelized memory network for video object segmentation. In *ECCV*, 2020. 2

[46] Xingjian SHI, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun WOO. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NeurIPS*, 2015. 3

[47] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014. 3

[48] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin Man Lam. Pyramid dilated deeper convlstm for video salient object detection. In *ECCV*, 2018. 7, 8

[49] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 6

[50] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning motion patterns in videos. In *CVPR*, 2017. 2, 3, 7

[51] P. Tokmakov, K. Alahari, and C. Schmid. Learning video object segmentation with visual memory. In *ICCV*, 2017. 2, 3, 7

[52] Carles Ventura, Miriam Bellver, Andreu Girbau, Amaia Salvador, Ferran Marques, and Xavier Giro-i Nieto. Rvos: End-to-end recurrent network for video object segmentation. In *CVPR*, 2019. 2

[53] Runzhong Wang, Junchi Yan, and Xiaokang Yang. Combinatorial learning of robust deep graph matching: an embedding based approach. *TPAMI*, 2020. 4, 5

[54] Wenguan Wang, Xiankai Lu, Jianbing Shen, David J. Crandall, and Ling Shao. Zero-shot video object segmentation via attentive graph neural networks. In *ICCV*, 2019. 3, 7

[55] Wenguan Wang, Jianbing Shen, and Ling Shao. Consistent video saliency using local gradient flow optimization and global refinement. *TIP*, 2015. 6, 7

[56] Wenguan Wang, Jianbing Shen, and Ling Shao. Video salient object detection via fully convolutional networks. *TIP*, 2017. 8

[57] Wenguan Wang, Hongmei Song, Shuyang Zhao, Jianbing Shen, Sanyuan Zhao, Steven CH Hoi, and Haibin Ling. Learning unsupervised video object segmentation through visual attention. In *CVPR*, 2019. 1, 7

[58] Huaxin Xiao, Jiashi Feng, Guosheng Lin, Yu Liu, and Maojun Zhang. Monet: Deep motion exploitation for video object segmentation. In *CVPR*, 2018. 2, 3

[59] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *ECCV*, 2018. 1, 2, 6, 7

[60] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, 2019. 1

[61] Zhao Yang, Qiang Wang, Luca Bertinetto, Weiming Hu, Song Bai, and Philip HS Torr. Anchor diffusion for unsupervised video object segmentation. In *ICCV*, 2019. 3, 6, 7, 8

[62] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. In *ECCV*, 2020. 2, 6

[63] A. Zanfir and C. Sminchisescu. Deep learning of graph matching. *CVPR*, 2018. 4, 5

[64] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *CVPR*, 2020. 4

[65] Ting Zhang, Guo-Jun Qi, Bin Xiao, and Jingdong Wang. Interleaved group convolutions. In *ICCV*, 2017. 3

[66] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, 2018. 3

[67] Yizhuo Zhang, Zhirong Wu, Houwen Peng, and Stephen Lin. A transductive approach for video object segmentation. In *CVPR*, 2020. 2, 3

[68] Mingmin Zhen, Shiwei Li, Lei Zhou, Jiaxiang Shang, Haoan Feng, Tian Fang, and Long Quan. Learning discriminative feature with crf for unsupervised video object segmentation. In *ECCV*, 2020. 3, 6, 7, 8

[69] Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *CVPR*, 2018. 3

[70] Tianfei Zhou, Shunzhou Wang, Yi Zhou, Yazhou Yao, Jianwu Li, and Ling Shao. Motion-attentive transition for zero-shot video object segmentation. In *AAAI*, 2020. 2, 3, 6, 7