Mining Relations among Cross-Frame Affinities for Video Semantic Segmentation

Guolei Sun[1], Yun Liu[1⋆], Hao Tang[1], Ajad Chhatkuli[1], Le Zhang[2], and Luc Van Gool[1,3]

[1] Computer Vision Lab, ETH Zurich
[2] School of Information and Communication Engineering, UESTC
[3] VISICS, KU Leuven

Abstract. The essence of video semantic segmentation (VSS) is how to leverage temporal information for prediction. Previous efforts are mainly devoted to developing new techniques to calculate the cross-frame affinities such as optical flow and attention. Instead, this paper contributes from a different angle by mining relations among cross-frame affinities, upon which better temporal information aggregation could be achieved. We explore relations among affinities in two aspects: single-scale intrinsic correlations and multi-scale relations. Inspired by traditional feature processing, we propose Single-scale Affinity Refinement (SAR) and Multi-scale Affinity Aggregation (MAA). To make it feasible to execute MAA, we propose a Selective Token Masking (STM) strategy to select a subset of consistent reference tokens for different scales when calculating affinities, which also improves the efficiency of our method. At last, the cross-frame affinities strengthened by SAR and MAA are adopted for adaptively aggregating temporal information. Our experiments demonstrate that the proposed method performs favorably against state-of-the-art VSS methods. The code is publicly available at https://github.com/GuoleiSun/VSS-MRCFA.

Keywords: Video semantic segmentation; Cross-frame affinities; Single-scale affinity refinement; Multi-scale affinity aggregation

1 Introduction

Image semantic segmentation aims at classifying each pixel of the input image to one of the predefined class labels, which is one of the most fundamental tasks in visual intelligence. Deep neural networks have made tremendous progresses in this field [41,52,5,21,55,17,18,58,30,50,24,25,11,10], benefiting from the availability of large-scale image datasets [9,54,3,35] for semantic segmentation. However, in real life, we usually confront more complex scenarios in which a series of successive video frames need to be segmented. Thus, it is desirable to explore video semantic segmentation (VSS) by exploiting the temporal information.

⋆ The corresponding author: yun.liu@vision.ee.ethz.ch

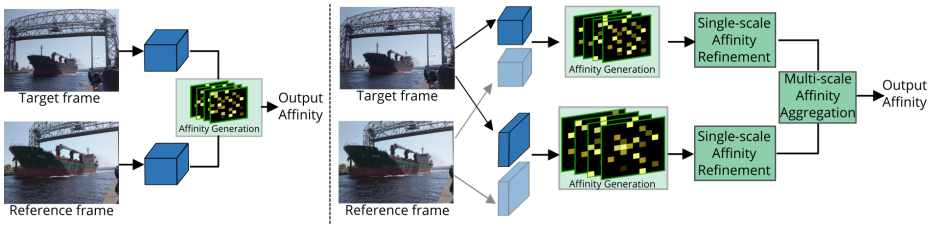

Fig. 1. *Left*: recent VSS methods [37,29] for which the affinity is directly forwarded to the next step (feature retrieval). The affinity is shown in a series of 2D maps. *Right*: We propose to mine the relations within the affinities before outputting the affinity, by Single-scale Affinity Refinement (SAR) and Multi-scale Affinity Aggregation (MAA).

The core of VSS is how to leverage temporal information. Most of the existing VSS works rely on the optical flow to model the temporal information. Specifically, they first compute the optical flow [14] that is further used to warp the features from neighboring video frames for feature alignment [56,16,48,36,22,33,28]. Then, the warped features can be simply aggregated. Although workable in certain scenarios, those methods are still unsatisfactory because i) the optical flow is error-prone and thus the error could be accumulated; ii) directly warping features may yield inevitable loss on the spatial correlations [31,20]. Hence, other approaches [37,29] directly aggregate the temporal information in the feature level using attention techniques, as shown in Fig. 1. Since they are conceptually simple and avoid the problems incurred by optical flow, we follow this way to exploit temporal information. In general, those methods first calculate the attentions/affinities between the target and the references, which are then used to generate the refined features. Though promising, they only consider the single-scale attention. What's more, they do not mine the relations within the affinities.

In this paper, we propose a novel approach MRCFA by Mining Relations among Cross-Frame Affinities for VSS. Specifically, we compute the **Cross-Frame Affinities (CFA)** between the features of the *target* frame and the *reference* frame. Hence, CFA is expected to have large activation for informative features and small activation for useless features. When aggregating the CFA-based temporal features, the informative features are highlighted and useless features are suppressed. As a result, the segmentation of the target frame would be improved by embedding temporal contexts. With the above analysis, the main focus of this paper is mining relations among CFA to improve the representation capability of CFA. Since deep neural networks usually generate multi-scale features and CFA can be calculated at different scales, we can obtain multi-scale CFA accordingly. Intuitively, the relations among CFA are twofold: single-scale intrinsic correlations and multi-scale relations.

For the *single-scale intrinsic correlations*, each feature token in a reference frame (*i.e.*, reference token) corresponds to a CFA map for the target frame. Intuitively, we have the observation that the CFA map of each reference token should be locally correlated as the feature map of the target frame is locally

correlated, which is also the basis of CNNs. It is interesting to note that the traditional 2D convolution can be adopted to model such single-scale intrinsic correlations of CFA. Generally, convolution is used for processing features. In contrast, we use convolution to refine the affinities of features for improving the quality of affinities. We call this step Single-scale Affinity Refinement (SAR).

For the *multi-scale relations*, we propose to exploit the relations among multi-scale CFA maps. The CFA maps generated from high-level features have a small scale and a coarse representation, while the CFA maps generated from low-level features have a large scale and a fine representation. It is natural to aggregate multi-scale CFA maps using a high-to-low decoder structure so that the resulting CFA would contain both coarse and fine affinities. Generally, the decoder structure is usually used for fusing multi-scale features. In contrast, we build a decoder to aggregate the multi-scale affinities of features. We call this step Multi-scale Affinity Aggregation (MAA).

When we revisit the above MAA, one requirement arises: the reference tokens at different scales should have the same number and corresponding semantics; otherwise, it is impossible to connect a decoder. As discussed above, each reference token corresponds to a CFA map for the target frame. Only when two reference tokens have the same semantics, their CFA maps can be merged. For this goal, a simple solution is to downsample reference tokens at different scales into the same size. This also saves the computation due to the reduction of reference tokens. It inspires us to further reduce the computation by sampling reference tokens. To this end, we propose a **Selective Token Masking** strategy to select S most important reference tokens and abandon less important ones. Then, the relation mining among CFA is executed based on the selected tokens.

In summary, there are three aspects for mining relations among CFA: i) We propose Single-scale Affinity Refinement for refining the affinities among features, based on single-scale intrinsic correlations; 2) We further introduce Multi-scale Affinity Aggregation by using an affinity decoder for aggregating the multi-scale affinities among features; 3) To make it feasible to execute MAA and improve efficiency, we propose Selective Token Masking (STM) to generate a subset of consistent reference tokens for each scale. After strengthened with single-scale and multi-scale relations, the final CFA can be directly used for embedding reference features into the target frame. Extensive experiments show the superiority of our method over previous VSS methods. Besides, our exploration of affinities among features would provide a new perspective on VSS.

2 Related Works

2.1 Image Semantic Segmentation

Image semantic segmentation has always been a hot topic in image understanding since it plays an important role in many real applications such as autonomous driving, robotic perception, augmented reality, aerial image analysis, and medical image analysis. In the era of deep learning, various algorithms have been proposed to improve semantic segmentation. Those related works can be divided into

two groups: CNN-based methods [41,51,8,19,5,49,40,44,1,12] and transformer-based methods [53,47]. Among CNN-based methods, FCN [41] is a pioneer work, which adopts fully convolutional networks and pixel-to-pixel classification. Since then, other methods [4,5,52,21,58,15] have been proposed to increase the receptive fields or representation ability of the network. Another group of works [53,47] is based on the transformer which is first proposed in natural language processing [45] and has the ability to capture global context [13]. Though tremendous progress has been achieved in image segmentation, researchers have paid more and more attention to VSS since video streams are a more realistic data modality.

2.2 Video Semantic Segmentation

Video semantic segmentation (VSS), aiming at classifying each pixel in each frame of a video into a predefined category, can be tackled by applying single image semantic segmentation algorithms [5,52,47,6,7] on each video frame. Though simple, this approach serves as an important baseline in VSS. One obvious drawback of this method is that the temporal information between consecutive frames is discarded and unexploited. Hence, dedicated VSS approaches [27,42,32,23,16,36,46,48,22,31,57,33,20,34,37,38,28] are proposed to make use of the temporal dimension to segment videos.

Most of the current VSS approaches can be divided into two groups. The first group of approaches focuses on using temporal information to reduce computation. Specifically, LLVS [31], Accel [22], GSVNET [28] and EVS [38] conserve computation by propagating the features from the key frames to non-key frames. Similarly, DVSNet [17] divides the current frame into different regions and the regions which do not differ much from previous frames do not traverse the slow segmentation network, but a fast flow network. However, due to the fact that they save computation on some frames or regions, their performance is usually inferior to the single frame baseline. The second group of methods focuses on exploring temporal information to improve segmentation performance and prediction consistency across frames. Specifically, NetWarp [46] wraps the features of the reference frames for temporal aggregation. TDNet [20] aggregates the features of sequential frames with an attention propagation module. ETC [33] uses motion information to impose temporal consistency among predictions between sequential frames. STT [29], LMANet [37] and CFFM [43] exploit the features from reference frames to help segment the target frame by the attention mechanism. Despite the promising results, those methods do not consider correlation mining among cross-frame affinities. This paper provides a new perspective on VSS by mining the relations among affinities.

3 Methodology

In this section, we target VSS and present a novel approach MRCFA through Mining Relations among Cross-Frame Affinities. The main idea of MRCFA is to mine the relations among multi-scale affinities computed from multi-scale

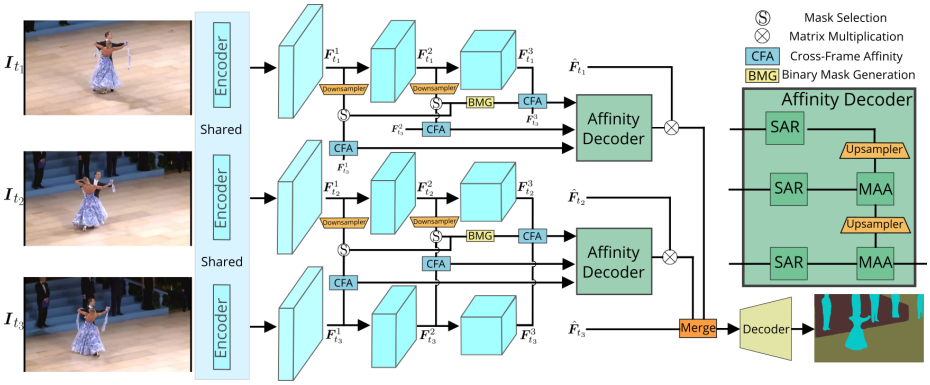

Fig. 2. Network overview of MRCFA. Our method is illustrated when the clip contains three frames ($T = 3$). The first two frames are reference frames while the last one is the target frame. All frames first go through the encoder to extract the multi-scale features ($L = 3$) from the intermediate layers. For each reference frame, we compute the Cross-Frame Affinities (CFA) across different scales of features. To save computation, Selective Token Masking is proposed. Then, the multi-scale affinities are input to an affinity decoder to learn a unified and informative affinity, through the Single-scale Affinity Refinement (SAR) module and Multi-scale Affinity Aggregation (MAA). The new representation of the target frame using the reference is obtained by exploiting the refined affinity to retrieve the corresponding reference features. Finally, all the new representations of the target are merged to segment the target. *Best viewed in color.*

intermediate features between the target frame and the reference frames, as illustrated in Fig. 2. We first provide the preliminaries in §3.1. Next, we introduce Single-scale Affinity Refinement (SAR) which independently refines each single-scale affinity in §3.2. After that, Multi-scare Affinities Aggregation (MAA) which merges affinities across various scales is presented in §3.3. Finally, we explain the Selective Token Masking mechanism (§3.4) to reduce the computation.

3.1 Preliminaries

Given a video clip $\{\boldsymbol{I}_{t_i} \in \mathbb{R}^{H\times W\times 3}\}_{i=1}^{T}$ containing T video frames and corresponding ground-truth masks $\{\boldsymbol{M}_{t_i} \in \mathbb{R}^{H\times W}\}_{i=1}^{T}$, our objective is to learn a VSS model. Without loss of generalizability, we focus on segmenting the last frame $\boldsymbol{I}_{t_T}$, which is referred as the target frame. All the previous frames $\{\boldsymbol{I}_{t_i}\}_{i=1}^{T-1}$ are referred as the reference frames. Each frame $\boldsymbol{I}_{t_i}$ is first input into an encoder to extract intermediate features $\{\boldsymbol{F}_{t_i}^{l} \in \mathbb{R}^{H_l W_l\times C_l}\}_{l=1}^{L}$ in various scales from L intermediate layers of the deep encoder, where H_l, W_l, C_l correspond to the height, width, number of channels of the feature map, respectively. For simplicity, multi-scale features $\{\boldsymbol{F}_{t_i}^{l}\}_{l=1}^{L}$ are in the order that shallow features are followed by deep features. We have $H_{l_1} \geq H_{l_2}$ and $W_{l_1} \geq W_{l_2}$, if $l_1 < l_2$. In this paper, we aim to exploit the contextual information in the reference frames to refine the features of the target frame and thus improve the target's segmentation.

Instead of simply modeling the affinities among frames for feature aggregation, we devote our efforts to mine relations among cross-frame affinities.

3.2 Single-scale Affinity Refinement

We start with introducing the process of generating multi-scale affinities between the target frame and each reference frame. We first map the features $\{\boldsymbol{F}_{t_T}^l\}_{l=1}^L$ of the target frames into the queries $\{\boldsymbol{Q}^l\}_{l=1}^L$ by a linear layer, as:

$$\boldsymbol{Q}^l = f(\boldsymbol{F}_{t_T}^l; \boldsymbol{W}_{query}^l), \tag{1}$$

where $\boldsymbol{W}_{query}^l \in \mathbb{R}^{C_l \times C_l}$ is the weight matrix of the linear layer f and $\boldsymbol{Q}^l \in \mathbb{R}^{H_l W_l \times C_l}$. Similarly, the multi-scale features $\{\boldsymbol{F}_{t_i}^l\}_{l=1}^L$ of the reference frame ($i \in [1, T-1]$) are also processed to generate the keys $\{\boldsymbol{K}_{t_i}^l\}_{l=1}^L$, as follows:

$$\boldsymbol{K}_{t_i}^l = f(\boldsymbol{F}_{t_i}^l; \boldsymbol{W}_{key}^l), \tag{2}$$

where $\boldsymbol{W}_{key}^l \in \mathbb{R}^{C_l \times C_l}$ is the corresponding weight matrix and $\boldsymbol{K}_{t_i}^l \in \mathbb{R}^{H_l W_l \times C_l}$. After obtaining the queries and the keys, we are ready to generate the affinities between the target frame $\boldsymbol{I}_{t_T}$ and each reference frame $\boldsymbol{I}_{t_i}$ ($i \in [1, T-1]$) across all scales. Then, **Cross-Frame Affinities (CFA)** are computed as:

$$\boldsymbol{A}_{t_i}^l = \boldsymbol{Q}^l \times \boldsymbol{K}_{t_i}^{l\top}, \tag{3}$$

where we have $\boldsymbol{A}_{t_i}^l \in \mathbb{R}^{H_l W_l \times H_l W_l}$, $l \in [1, L]$ and $i \in [1, T-1]$. It means that, at each scale, the target frame has an affinity map with each reference frame.

Based on the affinities $\{\boldsymbol{A}_{t_i}^l\}_{l=1}^L$, our affinity decoder is designed to mine the correlations between them to learn a better affinity between the target and the reference frame. As shown in Fig. 2, it is comprised of two modules: Single-scale Affinity Refinement (SAR) and Multi-scale Affinity Aggregation (MAA). Please refer to §1 for our motivations. In order to reduce computation and prepare the affinities for MAA module which requires the same number and corresponding semantics (see §1), our affinity decoder operates on $\{\tilde{\boldsymbol{A}}_{t_i}^l \in \mathbb{R}^{H_l W_l \times S}\}_{l=1}^L$, rather than $\{\boldsymbol{A}_{t_i}^l \in \mathbb{R}^{H_l W_l \times H_l W_l}\}_{l=1}^L$. The affinities $\tilde{\boldsymbol{A}}_{t_i}^l$ is a downsampled version of $\boldsymbol{A}_{t_i}^l$ along the second dimension, which will be explained in §3.4.

Single-scale Affinity Refinement (SAR). For the affinity matrix $\tilde{\boldsymbol{A}}_{t_i}^l$, each of its elements corresponds to a similarity between a token in the query and a token in the key. We reshape $\tilde{\boldsymbol{A}}_{t_i}^l$ from $\mathbb{R}^{H_l W_l \times S}$ to $\mathbb{R}^{H_l \times W_l \times S}$. In order to learn the correlation within the single-scale affinity $\tilde{\boldsymbol{A}}_{t_i}^l \in \mathbb{R}^{H_l \times W_l \times S}$, a straightforward way is to exploit 3D convolution. However, this approach suffers from two weaknesses. First, it requires a large amount of computational cost. Second, not all the activations within the 3D window are meaningful. Considering a 3D convolution with a kernel $\mathcal{K} \in \mathbb{R}^{k \times k \times k}$, the normal 3D convolution at the location $x = (x_1, x_2, x_3)$ is formulated as:

$$(\tilde{\boldsymbol{A}}_{t_i}^l * \mathcal{K})_x = \sum_{(o_1, o_2, o_3) \in \mathcal{N}(x)} \tilde{\boldsymbol{A}}_{t_i}^l(o_1, o_2, o_3) \mathcal{K}(o_1 - x_1, o_2 - x_2, o_3 - x_3), \tag{4}$$

where $\mathcal{N}(x)$ is the set of locations in the 3D window ($k \times k \times k$) centered at x, and $|\mathcal{N}(x)| = k^3$. As seen in Eq. (4), all the neighbors along three dimensions are used to conduct the 3D convolution. However, the last dimension of $\tilde{\boldsymbol{A}}^l_{t_i}$ is the sparse selection in the key (§3.4) and thus does not contain spatial information. Including the neighbors along the last dimension could introduce noise and bring more complexity. Thus, we propose to refine the affinities across the first two dimension. For affinity $\tilde{\boldsymbol{A}}^l_{t_i}$ of each scale, we first permute it to $\mathbb{R}^{S\times H_l\times W_l}$ and then use 2D convolutions to learn the relations within the affinity. The refined affinity is denoted as $\bar{\boldsymbol{A}}^l_{t_i} \in \mathbb{R}^{S\times H_l\times W_l}$. This process can be formulated as:

$$\begin{aligned} &\tilde{\boldsymbol{A}}^l_{t_i} \in \mathbb{R}^{H_l\times W_l\times S} \rightarrow \tilde{\boldsymbol{A}}^l_{t_i} \in \mathbb{R}^{S\times H_l\times W_l}, \\ &\bar{\boldsymbol{A}}^l_{t_i} = G(\tilde{\boldsymbol{A}}^l_{t_i}), \end{aligned} \tag{5}$$

where G represents a few connvolutional layers. Due to the use of 2D convolution and the token reduction mentioned in §3.4, the refinement of affinities is fast. After refining affinity for each scale, we collect the refined affinities $\{\bar{\boldsymbol{A}}^l_{t_i}\}^L_{l=1}$ for all scales. Next, we present Multi-scale Affinity Aggregation (MAA) module.

3.3 Multi-scale Affinity Aggregation

Multi-scale Affinity Aggregation (MAA). The affinity from the deep features contains more semantic but more coarse information, while the affinity from the shallow features contains more fine-grained but less semantic information. Thus, we propose a **Multi-scale Affinity Aggregation** module to aggregate the information from small-scale affinities to large-scale affinities, as:

$$\begin{aligned} &\boldsymbol{B}^L_{t_i} = \bar{\boldsymbol{A}}^L_{t_i}, \\ &\boldsymbol{B}^l_{t_i} = G(\Gamma(\boldsymbol{B}^{l+1}_{t_i}) + \bar{\boldsymbol{A}}^l_{t_i}), \quad l = L-1, ..., 1, \end{aligned} \tag{6}$$

where Γ denotes upsampling operation to match the spatial size when necessary. By Eq. (6), we generate the final refined affinity $\boldsymbol{B}^1_{t_i}$ between the target frame $\boldsymbol{I}_{t_T}$ and each reference frame $\boldsymbol{I}_{t_i}$ ($i \in [1, L-1]$).

Feature Retrieval. For single-frame semantic segmentation, SegFormer [47] generates the final feature $\hat{\boldsymbol{F}}_{t_i} \in \mathbb{R}^{\hat{H}\hat{W}\times\hat{C}}$ by merging multiple intermediate features. The final features are informative and directly used to predict the segmentation mask [47]. Using the refined affinity $\boldsymbol{B}^1_{t_i}$ and the informative features $\hat{\boldsymbol{F}}_{t_i}$, we compute the new refined feature representations for the target frame. Specifically, the feature $\hat{\boldsymbol{F}}_{t_i}$ is first downsampled to the size of $\mathbb{R}^{H_L W_L\times\hat{C}}$. To correspond the refined affinity and the informative feature, we sample feature $\hat{\boldsymbol{F}}_{t_i}$ using the token selection mask $\tilde{\boldsymbol{M}}_{t_i}$ (§3.4) and obtain $\tilde{\boldsymbol{F}}_{t_i} \in \mathbb{R}^{S\times\hat{C}}$. The new feature representation for the target frame using the reference is obtained as:

$$\boldsymbol{B}^1_{t_i} \in \mathbb{R}^{S\times H_1\times W_1} \rightarrow \boldsymbol{B}^1_{t_i} \in \mathbb{R}^{H_1 W_1\times S}, \qquad \boldsymbol{O}_{t_i} = \boldsymbol{B}^1_{t_i} \times \tilde{\boldsymbol{F}}_{t_i}. \tag{7}$$

Intuitively, this step is to retrieve the informative features from the reference frame to the target frame using affinity. Computing Eq. (7) for all reference frames, we obtain the new representations of the target frame as $\{\boldsymbol{O}_{t_i}\}^{T-1}_{i=0}$.

The final feature used to segment the target frame is merged from $\{\boldsymbol{O}_{t_i}\}_{i=0}^{T-1}$ and $\hat{\boldsymbol{F}}_{t_L}$ as follows:

$$\boldsymbol{O}_{t_L} = \frac{1}{T-1}\Gamma(\sum_{i=1}^{T-1}\boldsymbol{O}_{\boldsymbol{t_i}}) + \hat{\boldsymbol{F}}_{t_L}. \tag{8}$$

Finally, a simple MLP decoder projects $\boldsymbol{O}_{t_L}$ to the segmentation logits, and typical cross-entropy loss is used for training. In the test period, when segmenting the target frame I_{t_T}, the encoder only needs to generate the features for the current target while the reference frames are already processed in previous steps and the corresponding features can be directly used.

3.4 Selective Token Masking

As discussed in §1, there should be the same number of reference tokens with corresponding semantics across scales. Besides, computing cross-frame affinities requires a lot of computation. Thus, our affinity decoder does not process $\{\boldsymbol{A}_{t_i}^l \in \mathbb{R}^{H_l W_l \times H_l W_l}\}_{l=1}^L$, but rather its downsampled version $\{\tilde{\boldsymbol{A}}_{t_i}^l \in \mathbb{R}^{H_l W_l \times S}\}_{l=1}^L$. Here, we explain how to generate $\{\tilde{\boldsymbol{A}}_{t_i}^l\}_{l=1}^L$, by reducing the number of tokens in the multi-scale keys $\{\boldsymbol{K}_{t_i}^l\}_{l=1}^L$ before computing Eq. (3).

We exploit convolutional layers to downsample the multi-scale keys to the spatial size of $H_L \times W_L$. Specifically, for the key $\boldsymbol{K}_{t_i}^l$ ($l \in [1, L-1]$), we process it by a convolutional layer with both kernel and stride size of $(\frac{H_l}{H_L}, \frac{W_l}{W_L})$. As a result, we obtain new keys $\hat{\boldsymbol{K}}_{t_i}^l$ with smaller spatial size, which is given by

$$\begin{aligned} \boldsymbol{K}_{t_i}^l \in \mathbb{R}^{H_l W_l \times C_l} &\rightarrow \boldsymbol{K}_{t_i}^l \in \mathbb{R}^{C_l \times H_l \times W_l}, \\ \hat{\boldsymbol{K}}_{t_i}^l = g(\boldsymbol{K}_{t_i}^l; (\frac{H_l}{H_L}, \frac{W_l}{W_L}); (\frac{H_l}{H_L}, \frac{W_l}{W_L})), \\ \hat{\boldsymbol{K}}_{t_i}^l \in \mathbb{R}^{C_l \times H_L \times W_L} &\rightarrow \hat{\boldsymbol{K}}_{t_i}^l \in \mathbb{R}^{H_L W_L \times C_l}. \end{aligned} \tag{9}$$

where $g(\cdot; (k_h, k_w); (s_h, s_w))$ represents a convolutional layer with the kernel size (k_h, k_w) and the stride (s_h, s_w). After this step, we obtain the downsampled keys $\{\hat{\boldsymbol{K}}_{t_i}^l\}_{l=1}^{L-1}$, where $\hat{\boldsymbol{K}}_{t_i}^l \in \mathbb{R}^{H_L W_L \times C_l}$, $l \in [1, L-1]$ and $i \in [1, T-1]$.

To further reduce the number of tokens in $\{\hat{\boldsymbol{K}}_{t_i}^l\}_{l=1}^{L-1}$, we propose to select important tokens and discard less important ones. The idea is to first compute the affinity for the deepest query/key pair ($\boldsymbol{Q}^L$ and $\boldsymbol{K}_{t_i}^L$), then generate a binary mask of important token locations, and finally select tokens in keys using the mask. The process of **Binary Mask Generation (BMG)** is in the following. The affinity between the deepest query and key is given by $\boldsymbol{A}_{t_i}^L \in \mathbb{R}^{H_L W_L \times H_L W_L}$, following Eq. (3). Next, we choose the top-n maximum elements across each column of $\boldsymbol{A}_{t_i}^L$, given by

$$\hat{\boldsymbol{A}}_{t_i}^L[:, j] = \arg\max_n(\boldsymbol{A}_{t_i}^L[:, j]), \qquad j \in [1, H_L W_L], \tag{10}$$

where $\arg\max_n$ means to take the top-n elements, and $\hat{\boldsymbol{A}}_{t_i}^L \in \mathbb{R}^{n \times H_L W_L}$. Then, we sum over the top-n elements and generate a token importance map $\boldsymbol{M}_{t_i}$ as

$$\boldsymbol{M}_{t_i} = \sum_{j=1}^{n} (\hat{\boldsymbol{A}}_{t_i}^L[j, :]), \tag{11}$$

in which we have $\boldsymbol{M}_{t_i} \in \mathbb{R}^{H_L W_L}$. We recover the spatial size of $\boldsymbol{M}_{t_i}$ by reshaping it to $\mathbb{R}^{H_L \times W_L}$. The token importance map $\boldsymbol{M}_{t_i}$ shows the importance level of every location in the key feature map. Since $\boldsymbol{M}_{t_i}$ is derived from the deepest/highest level of features, the token importance information it contains is semantic-oriented and can be shared in other shallow levels. We use it to sample the tokens in $\{\hat{\boldsymbol{K}}_{t_i}^l\}_{l=1}^{L-1}$. Specifically, we sample p percent of the locations with the top-p highest importance scores in $\boldsymbol{M}_{t_i}$, where p is referred as the token selection ratio. The binary token selection mask with p percent of the locations highlighted is denoted as $\tilde{\boldsymbol{M}}_{t_i}$. The location with the value 1 in $\tilde{\boldsymbol{M}}_{t_i}$ means the token importance is within the top-p percent and the corresponding token will be selected. The location with the value 0 in $\tilde{\boldsymbol{M}}_{t_i}$ means the token in that location is less important and will thus be discarded. The total number of locations with the value 1 in $\tilde{\boldsymbol{M}}_{t_i}$ is denoted by $S = pH_L W_L$.

Using mask $\tilde{\boldsymbol{M}}_{t_i}$, we select p percent of tokens in $\{\hat{\boldsymbol{K}}_{t_i}^l\}_{l=1}^{L-1}$. The keys after selection are denoted as $\{\tilde{\boldsymbol{K}}_{t_i}^l \in \mathbb{R}^{S \times C_l}\}_{l=1}^{L-1}$. With $\boldsymbol{Q}^l$ and $\tilde{\boldsymbol{K}}_{t_i}^l$, we compute the affinities $\{\tilde{\boldsymbol{A}}_{t_i}^l \in \mathbb{R}^{H_l W_l \times S}\}_{l=1}^{L-1}$ using Eq. (3). For $\boldsymbol{A}_{t_i}^L$, we also conduct sampling using $\tilde{\boldsymbol{M}}_{t_i}$ and obtain $\tilde{\boldsymbol{A}}_{t_i}^L \in \mathbb{R}^{H_L W_L \times S}$. Merging the affinities from all L scales gives final affinities of $\{\tilde{\boldsymbol{A}}_{t_i}^l \in \mathbb{R}^{H_l W_l \times S}\}_{l=1}^{L}$. After computing the affinities for all reference frames, we have the downsampled affinities $\{\{\tilde{\boldsymbol{A}}_{t_i}^l\}_{l=1}^{L}\}_{i=1}^{T-1}$.

4 Experiments

4.1 Experimental Setup

Datasets. Densely annotating video frames requires intensive manual labeling efforts. The widely used datasets for VSS are Cityscapes [9] and CamVid [2] datasets. However, these datasets only contain sparse annotations, which limits the exploration of temporal information. Fortunately, the Video Scene Parsing in the Wild (VSPW) dataset [34] is proposed to facilitate the progress of this field. It is currently the largest-scale VSS dataset with 198,244 training frames, 24,502 validation frames and 28,887 test frames. For each video, 15 frames per second are densely annotated for 124 categories. These aspects make VSPW the best benchmark for VSS up till now. Hence, most of our experiments are conducted on VSPW. To further demonstrate the effectiveness of MRCFA, we also show results on Cityscapes, for which only one out of 30 frames is annotated.

Implementation details. For the encoder, we use the MiT backbones as in Segformer [47], which have been pretrained on ImageNet-1K [39]. For VSPW dataset, three reference frames are used, which are 9, 6 and 3 frames ahead of the

Methods	T	t_1	t_2	t_3	mIoU ↑	mVC_8 ↑	mVC_{16} ↑
SegFormer [47]	-	-	-	-	36.5	84.7	79.9
MRCFA (Ours)	2	-1	-	-	38.0	85.9	81.2
	2	-3	-	-	38.1	85.5	80.7
	2	-6	-	-	38.2	85.1	80.3
	2	-9	-	-	37.4	85.5	81.2
	3	-6	-3	-	38.4	87.0	82.1
	3	-9	-6	-	38.4	86.9	82.0
	4	-9	-6	-3	**38.9**	**88.8**	**84.4**

Table 1. The impact of the selection of reference frames.

p	mIoU ↑	mVC_8 ↑	mVC_{16} ↑	Memory (M) ↓	FPS (f/s) ↑
100%	39.4	89.2	84.9	1068	32.9
90%	39.1	89.1	84.8	1035	34.2
70%	39.1	88.2	83.9	969	36.8
50%	38.9	88.8	84.4	903 (15.4%)	40.1 (21.9%)
30%	38.5	86.7	81.9	838	43.5
10%	35.9	86.2	81.7	773	47.2

Table 2. The impact of token selection ratio p. The row which best deals with the trade-off between performance and computation resources is shown in red.

target, following [34]. Three-scale features from the last three transformer blocks are used to compute the cross-frame affinities and mine their correlations. For the Mask-based Token Selection (MTS), we set p=80% for MiT-B0 and p=50% for other backbones unless otherwise specified. For training augmentations, we use random resizing, horizontal flipping, and photometric distortion to process the original images. Then, the images are randomly cropped to the size of 480×480 to train the network. We set the batch size as 8 during training. The models are all trained with AdamW optimizer for a maximum of 160k iterations and "poly" learning rate schedule. The initial learning rate is 6e-5. For simplicity, we perform the single-scale test on the whole image, rather than the sliding window test or multi-scale test. The input images are resized to 480×853 for VSPW. We also do not perform any post-processing such as CRF [26]. For Cityscape, the input image is cropped to 512×1024 during training and resized to the same resolution during inference. And we use two reference frames and four-scale features. The number of frames being processed per second (FPS) is computed in a single Quadro RTX 6000 GPU (24G memory).

Evaluation metrics. To evaluate the segmentation results, we adopt the commonly used metrics of Mean IoU (mIoU) and Weighted IoU (WIoU), following [41]. We also use Video Consistency (VC) [34] to evaluate the category consistency among the adjacent frames in the video, following [34]. Formally, video consistency VC_n for n consecutive frames for a video clips $\{\boldsymbol{I}_c\}_{c=1}^{C}$, is computed by: $\text{VC}_n = \frac{1}{C-n+1} \sum_{i=1}^{C-n+1} \frac{(\cap_i^{i+n-1} \boldsymbol{S}_i) \cap (\cap_i^{i+n-1} \boldsymbol{S}_i')}{\cap_i^{i+n-1} \boldsymbol{S}_i}$, where $C \geq n$. $\boldsymbol{S}_i$ and $\boldsymbol{S}_i'$ are the ground-truth mask and predicted mask for i^{th} frame, respectively. We com-

pute the mean of video consistency VC_n for all videos in the dataset as mVC_n. Following [34], we compute mVC_8 and mVC_{16} to evaluate the visual consistency of the predicted masks. Please refer to [34] for more details about VC.

4.2 Ablation Studies

We conduct ablation studies on the large-scale VSPW dataset [34] to validate the key designs of MRCFA. For fairness, we adopt the same settings as in §4.1 unless otherwise specified. The ablation studies are conducted on MiT-B1 backbone.

Influence of the reference frames. We study the performance of our method with respect to different choices of reference frames in Tab. 1. We have the following observations. First, using a single reference frame largely improves the segmentation performance (mIoU). For example, when using a single reference frame which is 3 frames ahead of the target one, the mIoU improvement over the baseline (SegFormer) is 1.6%, *i.e.*, 38.1 over 36.5. Further adding more reference frames, better segmentation performance is observed. The best mIoU of 38.9 is obtained when using reference frames of 9, 6, and 3 frames ahead of the target. Second, for the prediction consistency metrics (mVC_8 and mVC_{16}), the advantage of exploiting more reference frames is more obvious. For example, using one reference frame ($t_1 = -6$) gives mVC_8 and mVC_{16} of 85.1 and 80.3, improving the baseline by 0.4% and 0.4%, respectively. However, when using three reference frames ($t_1 = -9$, $t_2 = -6$, $t_3 = -3$), the achieved mVC_8 and mVC_{16} are much more superior to the baseline, improving by 4.1% and 4.5%. The results are reasonable because using more reference frames gives the model a bigger view of the previously predicted features and thus generates more consistent predictions.

Influence of token selection ratio p. We study the influence of the token selection ratio p in terms of performance and computational resources in Tab. 2. Smaller p represents that less number of tokens in the key features are selected and thus less computation resource is required. Hence, there is a trade-off between the segmentation performance and the required resources (GPU memory and additional latency). In the experiments, when reducing $p = 100\%$ to 50%, the performance reduces slightly (0.5 in mIoU) while the GPU memory reduces by 15.4% and FPS increases by 21.9%. When further reducing p to 10%, the performance largely decreases in terms of mIoU, mVC_8 and mVC_{16}. The reason is that too many tokens are discarded in the reference frames and the remained tokens are not informative enough to provide the required contexts for segmenting the target frame. To sum up, the best trade-off is achieved when $p = 50\%$.

Influence of the feature scales. For VSPW dataset, we use three-scale features output from the last three transformer blocks. Here, we conduct an ablation study on the impact of the used feature scales. The results are shown in Tab. 3. It can be observed that using the features from the last stage ($L = 1$) or the last two stages ($L = 2$) gives inferior performance while consuming less computational resources and achieving faster running speed. When using three-scale features, the best results are achieved in terms of mIoU, mVC_8, and mVC_{16}. This

L	mIoU ↑	mVC_8 ↑	mVC_{16} ↑	Params (M) ↓	FPS (f/s) ↑
1	37.5	87.7	83.1	14.8	44.3
2	38.1	87.5	82.5	15.3	43.8
3	**38.9**	**88.8**	**84.4**	16.2	40.1

Table 3. Ablation study on the number of feature scales (L). Using more scales of features for our method progressively increases the performance.

Methods	SAR	MAA	mIoU ↑	mVC_8 ↑	mVC_{16} ↑	Params (M) ↓
SegFormer	-	-	36.5	84.7	79.9	13.8
Feature Pyramid	-	-	37.8	87.0	82.0	16.2
Affinity Decoder	✓	✗	37.8	87.1	82.6	16.2
	✗	✓	37.4	88.3	83.6	16.2
	✓	✓	**38.9**	**88.8**	**84.4**	16.2

Table 4. Ablation study on the affinity decoder. Within our design, SAR and MAA are essential parts which contribute to the refinement of the affinity.

is due to the fact that the features in different scales contain complementary information, and the proposed affinity decoder successfully mines this information through learning correlations between multi-scale affinities.

Ablation study on affinity decoder. We conduct ablation studies on the proposed affinity decoder. The results are shown in Tab. 4. Our affinity decoder processes the multi-scale affinities and generates a refined affinity matrix for each pair of the target and reference frames. It is reasonable to ask whether this design is better than the feature pyramid baseline. For this baseline (Feature Pyramid), we first compute the features for the target frame using the reference frame features at each scale and then merge those multi-scale features. For fair comparisons, we use a similar number of parameters for this baseline and other settings are also the same as ours. The result shows that while Feature Pyramid performs favorably over the single-frame baseline, our approach clearly surpasses it. It validates the effectiveness of the proposed affinity decoder.

As presented in §3.2, our affinity decoder has two modules: Single-scale Affinity Refinement (SAR) and Multi-scale Affinity Aggregation (MAA). The ablation study of two modules is provided in Tab. 4. Only using SAR, our method obtains the mIoU of 37.8, while only using MAA gives the mIoU of 37.4. Both variants are clearly better than the baseline, validating their effectiveness. Combining both modules, the proposed approach achieves the best mIoU, mVC_8, and mVC_{16}. It shows that both SAR and MAA are essential parts of the affinity decoder to learn better affinities to help segment the target frame.

4.3 Segmentation Results

The state-of-the-art comparisons on VSPW [34] dataset are shown in Tab. 5. Besides segmentation performance and visual consistency of the predicted masks,

Methods	Backbone	mIoU ↑	Weighted IoU ↑	mVC_8 ↑	mVC_{16} ↑	Params (M) ↓	FPS (f/s) ↑
SegFormer [47]	MiT-B0	32.9	56.8	82.7	77.3	3.8	73.4
SegFormer [47]	MiT-B1	36.5	58.8	84.7	79.9	13.8	58.7
MRCFA (Ours)	MiT-B0	35.2	57.9	88.0	83.2	5.2	50.0
MRCFA (Ours)	MiT-B1	**38.9**	**60.0**	**88.8**	**84.4**	16.2	40.1
DeepLabv3+ [6]	ResNet-101	34.7	58.8	83.2	78.2	62.7	-
UperNet [46]	ResNet-101	36.5	58.6	82.6	76.1	83.2	-
PSPNet [52]	ResNet-101	36.5	58.1	84.2	79.6	70.5	13.9
OCRNet [50]	ResNet-101	36.7	59.2	84.0	79.0	58.1	14.3
ETC [33]	PSPNet	36.6	58.3	84.1	79.2	89.4	-
NetWarp [46]	PSPNet	37.0	57.9	84.4	79.4	89.4	-
ETC [33]	OCRNet	37.5	59.1	84.1	79.1	58.1	-
NetWarp [46]	OCRNet	37.5	58.9	84.0	79.0	58.1	-
$TCB_{st\text{-}ppm}$ [34]	ResNet-101	37.5	58.6	87.0	82.1	70.5	10.0
$TCB_{st\text{-}ocr}$ [34]	ResNet-101	37.4	59.3	86.9	82.0	58.1	5.5
$TCB_{st\text{-}ocr\text{-}mem}$ [34]	ResNet-101	37.8	59.5	87.9	84.0	58.1	5.5
SegFormer [47]	MiT-B2	43.9	63.7	86.0	81.2	24.8	39.2
SegFormer [47]	MiT-B5	48.2	65.1	87.8	83.7	82.1	17.2
MRCFA (Ours)	MiT-B2	45.3	64.7	90.3	86.2	27.3	32.1
MRCFA (Ours)	MiT-B5	**49.9**	**66.0**	**90.9**	**87.4**	84.5	15.7

Table 5. State-of-the-art comparison on the VSPW [34] validation set. MRCFA outperforms the compared methods on both accuracy (mIoU) and prediction consistency.

we also report the model complexity and FPS. According to the model size, the methods are divided into two groups: small models and large models. Among all methods, our MRCFA achieves state-of-the-art performance and produces the most consistent segmentation masks across video frames. For small models, our method on MiT-B1 clearly outperforms the strong baseline SegFormer [47] by 2.4% in mIoU and 1.2% in weighted IoU. In terms of the visual consistency in the predicted masks, our approach is superior to other methods, surpassing the second best method with 4.1% and 4.5% in mVC_8 and mVC_{16}, respectively. For large models, MRCFA shows similar behavior. The results indicate that our method is effective in mining the relations between the target and reference frames through the designed modules: SAR and MAA.

Methods	Backbone	mIoU ↑	Params (M) ↓	FPS (f/s) ↑
FCN [41]	MobileNetV2	61.5	9.8	14.2
CC [42]	VGG-16	67.7	-	16.5
DFF [56]	ResNet-101	68.7	-	9.7
GRFP [36]	ResNet-101	69.4	-	3.2
PSPNet [52]	MobileNetV2	70.2	13.7	11.2
DVSN [48]	ResNet-101	70.3	-	19.8
Accel [22]	ResNet-101	72.1	-	3.6
ETC [33]	ResNet-18	71.1	13.2	9.5
SegFormer [47]	MiT-B0	71.9	3.7	58.5
MRCFA (Ours)	MiT-B0	72.8	4.2	33.3
SegFormer [47]	MiT-B1	74.1	13.8	46.8
MRCFA (Ours)	MiT-B1	**75.1**	14.9	21.5

Table 6. State-of-the-art comparison on the Cityscapes [9] val set.

Despite that our approach achieves impressive performance, it adds limited model complexity and latency. Specifically, compared to SegFormer (MiT-B2), MRCFA slightly increases the number of parameters from 24.8M to 27.3M and reduces the FPS from 39.2 to 32.1. The efficiency of our method benefits from the proposed STM mechanism for which we abandon unimportant tokens.

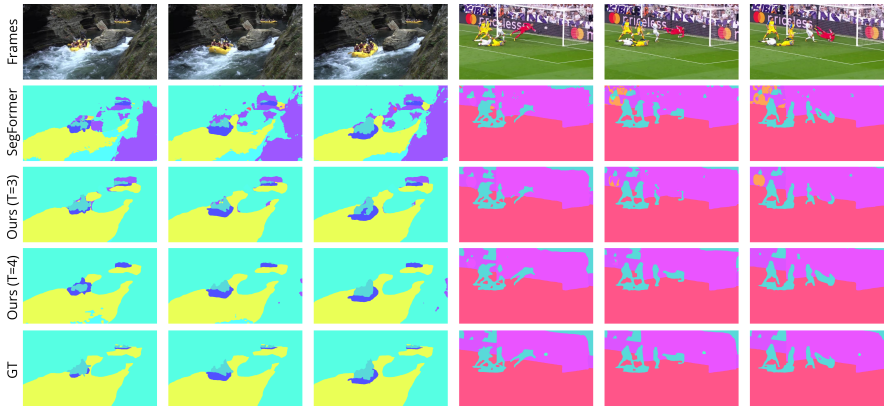

Fig. 3. Qualitative results. From *top* to *bottom*: the input frames, the predicted masks of SegFormer [47], the predictions of ours ($T = 3, t_1 = -3$, $t_2 = -6$), the predictions of ours ($T = 4, t_1 = -3$, $t_2 = -6$, $t_3 = -9$) and the ground-truth masks. Our model generates better results than the baseline in terms of accuracy and VC.

We conduct additional experiments on the semi-supervised Cityscapes [9] dataset, for which only one frame in each video clip is pixel-wise annotated. Tab. 6 shows the results. Similar to VSPW, MRCFA also achieves state-of-the-art results among the compared approaches under the semi-supervised setting and has a fast running speed. Besides the quantitative comparisons analyzed above, we also qualitatively compare the proposed method with the baseline on the sampled video clips in Fig. 3. For the two samples, our method generates more accurate segmentation masks, which are also more visually consistent.

5 Conclusions

This paper presents a novel framework MRCFA for VSS. Different from previous methods, we aim at mining the relations among multi-scale Cross-Frame Affinities (CFA) in two aspects: single-scale intrinsic correlations and multi-scale relations. Accordingly, Single-scale Affinity Refinement (SAR) is proposed to independently refine the affinity of each scale, while Multi-scale Affinity Aggregation (MAA) is designed to merge the refined affinities across various scales. To reduce computation and facilitate MAA, Selective Token Masking (STM) is adopted to sample important tokens in keys for the reference frames. Combining all the novelties, MRCFA generates better affinity relations between the target and the reference frames without largely adding computational resources. Extensive experiments demonstrate the effectiveness and efficiency of MRCFA, by setting new state-of-the-arts. The key components are validated to be essential for our method by ablation studies. Overall, our exploration of mining the relations among affinities could provide a new perspective on VSS.

References

1. Ahn, J., Cho, S., Kwak, S.: Weakly supervised learning of instance segmentation with inter-pixel relations. In: IEEE CVPR. pp. 2209–2218 (2019) 4
2. Brostow, G.J., Fauqueur, J., Cipolla, R.: Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters **30**(2), 88–97 (2009) 9
3. Caesar, H., Uijlings, J., Ferrari, V.: COCO-Stuff: Thing and stuff classes in context. In: IEEE CVPR. pp. 1209–1218 (2018) 1
4. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected CRFs. In: ICLR (2015) 4
5. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE TPAMI **40**(4), 834–848 (2018) 1, 4
6. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017) 4, 13
7. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV. pp. 801–818 (2018) 4
8. Chen, W., Zhu, X., Sun, R., He, J., Li, R., Shen, X., Yu, B.: Tensor low-rank reconstruction for semantic segmentation. In: ECCV. pp. 52–69 (2020) 4
9. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes dataset for semantic urban scene understanding. In: IEEE CVPR. pp. 3213–3223 (2016) 1, 9, 13, 14
10. Ding, H., Jiang, X., Liu, A.Q., Thalmann, N.M., Wang, G.: Boundary-aware feature propagation for scene segmentation. In: IEEE ICCV. pp. 6819–6829 (2019) 1
11. Ding, H., Jiang, X., Shuai, B., Liu, A.Q., Wang, G.: Context contrasted feature and gated multi-scale aggregation for scene segmentation. In: IEEE CVPR. pp. 2393–2402 (2018) 1
12. Ding, H., Jiang, X., Shuai, B., Liu, A.Q., Wang, G.: Semantic correlation promoted shape-variant context for segmentation. In: IEEE CVPR. pp. 8885–8894 (2019) 4
13. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021) 4
14. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: FlowNet: Learning optical flow with convolutional networks. In: IEEE ICCV. pp. 2758–2766 (2015) 2
15. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: IEEE CVPR. pp. 3146–3154 (2019) 4
16. Gadde, R., Jampani, V., Gehler, P.V.: Semantic video CNNs through representation warping. In: IEEE ICCV. pp. 4453–4462 (2017) 2, 4
17. He, J., Deng, Z., Qiao, Y.: Dynamic multi-scale filters for semantic segmentation. In: IEEE ICCV. pp. 3562–3572 (2019) 1, 4
18. He, J., Deng, Z., Zhou, L., Wang, Y., Qiao, Y.: Adaptive pyramid context network for semantic segmentation. In: IEEE CVPR. pp. 7519–7528 (2019) 1
19. Hsiao, C.W., Sun, C., Chen, H.T., Sun, M.: Specialize and fuse: Pyramidal output representation for semantic segmentation. In: IEEE ICCV. pp. 7137–7146 (2021) 4

20. Hu, P., Caba, F., Wang, O., Lin, Z., Sclaroff, S., Perazzi, F.: Temporally distributed networks for fast video semantic segmentation. In: IEEE CVPR. pp. 8818–8827 (2020) 2, 4
21. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: CCNet: Criss-cross attention for semantic segmentation. In: IEEE ICCV. pp. 603–612 (2019) 1, 4
22. Jain, S., Wang, X., Gonzalez, J.E.: Accel: A corrective fusion network for efficient semantic segmentation on video. In: IEEE CVPR. pp. 8866–8875 (2019) 2, 4, 13
23. Jin, X., Li, X., Xiao, H., Shen, X., Lin, Z., Yang, J., Chen, Y., Dong, J., Liu, L., Jie, Z., et al.: Video scene parsing with predictive feature learning. In: IEEE ICCV. pp. 5580–5588 (2017) 4
24. Jin, Z., Gong, T., Yu, D., Chu, Q., Wang, J., Wang, C., Shao, J.: Mining contextual information beyond image for semantic segmentation. In: IEEE ICCV. pp. 7231–7241 (2021) 1
25. Jin, Z., Liu, B., Chu, Q., Yu, N.: ISNet: Integrate image-level and semantic-level context for semantic segmentation. In: IEEE ICCV. pp. 7189–7198 (2021) 1
26. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected CRFs with Gaussian edge potentials. In: NeurIPS. pp. 109–117 (2011) 10
27. Kundu, A., Vineet, V., Koltun, V.: Feature space optimization for semantic video segmentation. In: IEEE CVPR. pp. 3168–3175 (2016) 4
28. Lee, S.P., Chen, S.C., Peng, W.H.: GSVNet: Guided spatially-varying convolution for fast semantic segmentation on video. In: IEEE ICME. pp. 1–6 (2021) 2, 4
29. Li, J., Wang, W., Chen, J., Niu, L., Si, J., Qian, C., Zhang, L.: Video semantic segmentation via sparse temporal transformer. In: ACM MM. pp. 59–68 (2021) 2, 4
30. Li, X., Yang, Y., Zhao, Q., Shen, T., Lin, Z., Liu, H.: Spatial pyramid based graph reasoning for semantic segmentation. In: IEEE CVPR. pp. 8950–8959 (2020) 1
31. Li, Y., Shi, J., Lin, D.: Low-latency video semantic segmentation. In: IEEE CVPR. pp. 5997–6005 (2018) 2, 4
32. Liu, S., Wang, C., Qian, R., Yu, H., Bao, R., Sun, Y.: Surveillance video parsing with single frame supervision. In: IEEE CVPR. pp. 413–421 (2017) 4
33. Liu, Y., Shen, C., Yu, C., Wang, J.: Efficient semantic video segmentation with per-frame inference. In: ECCV. pp. 352–368 (2020) 2, 4, 13
34. Miao, J., Wei, Y., Wu, Y., Liang, C., Li, G., Yang, Y.: VSPW: A large-scale dataset for video scene parsing in the wild. In: IEEE CVPR. pp. 4133–4143 (2021) 4, 9, 10, 11, 12, 13
35. Neuhold, G., Ollmann, T., Rota Bulo, S., Kontschieder, P.: The Mapillary Vistas dataset for semantic understanding of street scenes. In: IEEE ICCV. pp. 4990–4999 (2017) 1
36. Nilsson, D., Sminchisescu, C.: Semantic video segmentation by gated recurrent flow propagation. In: IEEE CVPR. pp. 6819–6828 (2018) 2, 4, 13
37. Paul, M., Danelljan, M., Van Gool, L., Timofte, R.: Local memory attention for fast video semantic segmentation. In: IROS. pp. 1102–1109. IEEE (2021) 2, 4
38. Paul, M., Mayer, C., Gool, L.V., Timofte, R.: Efficient video semantic segmentation with labels propagation and refinement. In: Winter Conf. Appli. Comput. Vis. (WACV). pp. 2873–2882 (2020) 4
39. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: ImageNet large scale visual recognition challenge. IJCV **115**(3), 211–252 (2015) 9
40. Seifi, S., Tuytelaars, T.: Attend and segment: Attention guided active semantic segmentation. In: ECCV. pp. 305–321 (2020) 4

41. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE TPAMI **39**(4), 640–651 (2017) 1, 4, 10, 13
42. Shelhamer, E., Rakelly, K., Hoffman, J., Darrell, T.: Clockwork convnets for video semantic segmentation. In: ECCV. pp. 852–868 (2016) 4, 13
43. Sun, G., Liu, Y., Ding, H., Probst, T., Van Gool, L.: Coarse-to-fine feature mining for video semantic segmentation. In: IEEE CVPR. pp. 3126–3137 (2022) 4
44. Sun, G., Wang, W., Dai, J., Van Gool, L.: Mining cross-image semantics for weakly supervised semantic segmentation. In: ECCV. pp. 347–365. Springer (2020) 4
45. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NeurIPS. pp. 6000–6010 (2017) 4
46. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: ECCV. pp. 418–434 (2018) 4, 13
47. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: SegFormer: Simple and efficient design for semantic segmentation with transformers. In: NeurIPS (2021) 4, 7, 9, 10, 13, 14
48. Xu, Y.S., Fu, T.J., Yang, H.K., Lee, C.Y.: Dynamic video segmentation network. In: IEEE CVPR. pp. 6556–6565 (2018) 2, 4, 13
49. Yang, M., Yu, K., Zhang, C., Li, Z., Yang, K.: DenseASPP for semantic segmentation in street scenes. In: IEEE CVPR. pp. 3684–3692 (2018) 4
50. Yuan, Y., Chen, X., Wang, J.: Object-contextual representations for semantic segmentation. In: ECCV. pp. 173–190 (2020) 1, 13
51. Zhang, F., Chen, Y., Li, Z., Hong, Z., Liu, J., Ma, F., Han, J., Ding, E.: ACFNet: Attentional class feature network for semantic segmentation. In: IEEE ICCV. pp. 6798–6807 (2019) 4
52. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: IEEE CVPR. pp. 2881–2890 (2017) 1, 4, 13
53. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., Zhang, L.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: IEEE CVPR. pp. 6881–6890 (2021) 4
54. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ADE20K dataset. IJCV **127**(3), 302–321 (2019) 1
55. Zhou, Y., Sun, X., Zha, Z.J., Zeng, W.: Context-reinforced semantic segmentation. In: IEEE CVPR. pp. 4046–4055 (2019) 1
56. Zhu, X., Xiong, Y., Dai, J., Yuan, L., Wei, Y.: Deep feature flow for video recognition. In: IEEE CVPR. pp. 2349–2358 (2017) 2, 13
57. Zhu, Y., Sapra, K., Reda, F.A., Shih, K.J., Newsam, S., Tao, A., Catanzaro, B.: Improving semantic segmentation via video propagation and label relaxation. In: IEEE CVPR. pp. 8856–8865 (2019) 4
58. Zhu, Z., Xu, M., Bai, S., Huang, T., Bai, X.: Asymmetric non-local neural networks for semantic segmentation. In: IEEE ICCV. pp. 593–602 (2019) 1, 4