Learning Local and Global Temporal Contexts for Video Semantic Segmentation

Guolei Sun, Yun Liu, Henghui Ding, Min Wu, and Luc Van Gool

Abstract—Contextual information plays a core role for video semantic segmentation (VSS). This paper summarizes contexts for VSS in two-fold: *local* temporal contexts (LTC) which define the contexts from neighboring frames, and *global* temporal contexts (GTC) which represent the contexts from the whole video. As for LTC, it includes static and motional contexts, corresponding to static and moving content in neighboring frames, respectively. Previously, both static and motional contexts have been studied. However, there is no research about simultaneously learning static and motional contexts (highly complementary). Hence, we propose a Coarse-to-Fine Feature Mining (CFFM) technique to learn a unified presentation of LTC. CFFM contains two parts: Coarse-to-Fine Feature Assembling (CFFA) and Cross-frame Feature Mining (CFM). CFFA abstracts static and motional contexts, and CFM mines useful information from nearby frames to enhance target features. To further exploit more temporal contexts, we propose CFFM++ by additionally learning GTC from the whole video. Specifically, we uniformly sample certain frames from the video and extract global contextual prototypes by k-means. The information within those prototypes is mined by CFM to refine target features. Experimental results on popular benchmarks demonstrate that CFFM and CFFM++ perform favorably against state-of-the-art methods. The code is available at https://github.com/GuoleiSun/VSS-CFFM.

Index Terms—Video semantic segmentation, local temporal contexts, static contexts, motional contexts, global temporal contexts, feature mining, vision transformer

1 Introduction

Semantic segmentation aims at assigning a semantic label to each pixel in a natural image, which is a fundamental and hot topic in the computer vision community. It has a wide range of applications in both academic and industrial fields. Thanks to the powerful representation capability of deep neural networks [2]–[5] and large-scale image datasets [6]–[10], tremendous achievements have been seen for image semantic segmentation. However, **video semantic segmentation (VSS)** has not been witnessed such tremendous progress [11]–[14] due to the lack of large-scale datasets. For example, Cityscapes [7] and NYUDv2 [15] datasets only annotate one or several nonadjacent frames in a video clip. CamVid [16] only has a small scale and a low frame rate. The real world is actually dynamic rather than static, so research on VSS is necessary. Fortunately, the recent establishment of the large-scale VSS dataset, VSPW [17], solves the problem of video data scarcity. This inspires us to denote our efforts to VSS.

As widely accepted, the contextual information plays a central role in image semantic segmentation [18]–[33]. When considering videos, the contextual information can be divided into two cases based on how much temporal information is used: ***local temporal contexts*** and ***global temporal contexts***. As shown in Fig. 1a, local temporal contexts refer to the contexts from neighboring/nearby frames, while global temporal contexts represent the contexts from a much larger view, *i.e.*, the whole video.

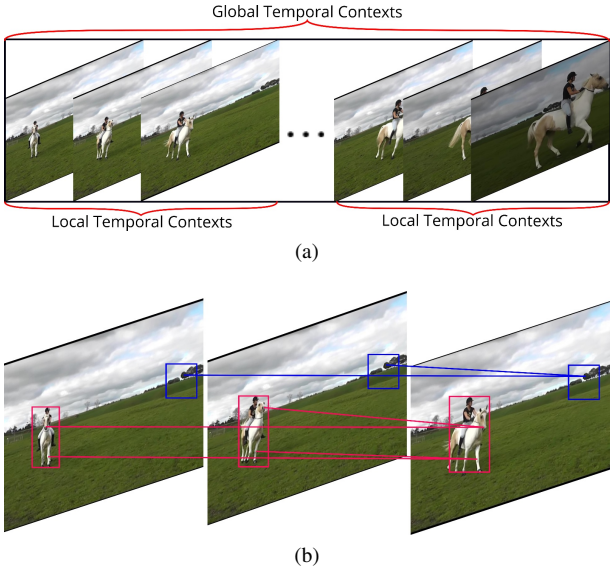


Fig. 1. **Illustration of various video contexts.** (a) Illustration of *local temporal contexts* and *global temporal contexts*. (b) Illustration of *static contexts* (in blue) and *motional contexts* (in red) across neighbouring video frames. The human and horse are moving objects, while the grassland and sky are static backgrounds. Note that the static stuff is helpful for the recognition of moving objects, *i.e.*, a human is riding a horse on the grassland.

We first discuss local temporal contexts which are widely exploited in VSS [11]–[14], [34]–[42]. The local temporal contexts can be further divided into ***static contexts*** and ***motional contexts*** among neighboring video frames, as shown in Fig. 1b. The former refers to the contexts within the same video frame or the contexts of unchanged content across the neighboring frames. Image semantic segmentation has exploited such contexts (for

- *G. Sun and L. V. Gool are with Computer Vision Lab, ETH Zurich, Zurich, Switzerland.*
- *Y. Liu and M. Wu are with the Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (A*STAR), Singapore.*
- *H. Ding is with MMLab@NTU, Nanyang Technological University, Singapore.*

- *A preliminary version of this work has been published on CVPR 2022 [1].*

- *Corresponding author: Yun Liu (E-mail: VAGRANTLYUN@GMAIL.COM)*

images) a lot, mainly accounting for multi-scale [24], [27], [28], [30] and global/long-range information [18], [29], [31], [32]. Such information is essential not only for understanding the static scene but also for perceiving the relatively holistic environment existing in the neighboring frames. The latter is responsible for better parsing moving objects/stuff and capturing more effective scene representations with the help of motions. Most of the VSS methods mainly studied motional contexts among nearby frames, which usually relies on optical flows [43] to model motional contexts from frames to adjacent frames, ignoring the static contexts. Although each single aspect, *i.e.*, static or motional contexts, has been well studied, how to learn static and motional contexts simultaneously among nearby frames deserves more attention, which is important for VSS.

Furthermore, static contexts and motional contexts are highly correlated, not isolated, because both contexts are complementary to each other to represent the information existing in several nearby frames. Therefore, the ideal solution for learning local temporal contexts is to jointly learn static and motional contexts, *i.e.*, generating a unified representation of static and motional contexts. A naïve solution is to apply recent popular self-attention [44]–[46] by taking feature vectors at all pixels in neighboring frames as tokens. This can directly model global relationships of all tokens, of course including both static and motional contexts. However, this naïve solution has some obvious drawbacks. For example, it is super inefficient due to a large number of tokens/pixels in the considered nearby frames, making this naïve solution unrealistic. It also contains too much redundant computation because most content in nearby frames usually does not change much and it is unnecessary to compute attention for the repeated content. Moreover, the too-long length of tokens would affect the performance of self-attention, as shown in [47]–[51] where the reduction of the token length through downsampling leads to better performance. More discussion about why traditional self-attention is inappropriate for video context learning can be found in §3.1.

In this paper, we propose a **Coarse-to-Fine Feature Mining (CFFM)** technique to learn local temporal contexts, which consists of two parts: **Coarse-to-Fine Feature Assembling (CFFA)** and **Cross-frame Feature Mining (CFM)**. Specifically, we first apply an efficient deep network [52] to extract features from each frame. Then, we assemble the extracted features from neighboring frames in a coarse-to-fine manner. Here, we use a larger receptive field and a more coarse pooling if the frame is more distant from the target frame. This feature assembling operation has two meanings. On one hand, it organizes the features in a multi-scale way, and the farthest frame would have the largest receptive field and the most coarse pooling. Since the content in a few sequential frames usually does not change suddenly and most content may only have a little temporal inconsistency, this operation is expected to prepare data for learning static contexts. On the other hand, this feature assembling operation enables a large perception region for remote frames because the moving objects may appear in a large region for remote frames. This makes it suitable for learning motional contexts. Then, with the assembled features, we use the CFM technique to iteratively mine useful contextual information from neighbouring frames for the target frame. This mining technique is a specially designed non-self attention mechanism that has two different inputs, unlike commonly used self-attention that only has one input [44], [45]. The output features enhanced by CFFM can be directly used for final prediction. We describe the technical motivations for CFFM in detail in §3.1.

For global temporal contexts, few VSS methods [17], [53] have exploited the contexts from the whole video. The modeling of global temporal contexts is usually achieved by a memory module in the form of a memory bank [17] or a tiny network [53] which is updated during inference. Although promising results have been achieved, there are two obvious drawbacks: 1) the global temporal contexts are *implicitly* modeled and it is unclear what information is kept in the memory; 2) the contextual information in the memory keeps increasing when processing the video frame-by-frame and the global temporal interaction is only possible for the last few frames of the video. To this end, based on the proposed CFFM, we further propose to *explicitly* learn global temporal contexts for VSS. After training our CFFM, the features (for each frame of the video) output from the trained network contains high-level semantic information and can be used to extract global temporal contexts. Since a video contains a large number of frames (tens or hundreds), we first sample some frames by a certain step from the whole video. This largely reduces the number of frames for the following processing. Features are extracted for these selected frames, which are decomposed as tokens. Here, the number of tokens is still large and impossible to be used. We largely reduce the token quantity by clustering all the tokens into different sub-groups. The centers of sub-groups are informative and representative prototypes, which abstract the contexts for the whole video. With the generated prototypes, we use the CFM technique again to iteratively mine useful information from the whole video to the target frame. The prediction of this global temporal context mining branch is combined with the prediction from CFFM. We name this model using both local and global temporal contexts as CFFM++, which is an extension of the CFFM by incorporating global temporal contexts.

To summarize, this paper studies local and global temporal contexts for VSS, with the following contributions:

- To learn the *local temporal contexts* of videos, we propose CFFM technique to learn a unified representation of *static contexts* and *motional contexts* among neighboring video frames, both of which are of vital importance for VSS.
- To learn the *global temporal contexts* of videos, we propose a global temporal context mining module to explicitly incorporate contextual information from the whole video to the target frame.
- Without bells and whistles, we achieve state-of-the-art results for VSS on standard benchmarks by using the CFFM technique. With the global temporal contexts incorporated, CFFM++ further boosts the performance of VSS.

We build this paper upon our recent conference paper [1] and significantly extend it in various ways. First, we propose an extension method CFFM++ (Fig. 3) based on the original framework (CFFM) to exploit *global temporal contexts* from the whole video, further boosting the segmentation performance while introducing only limited computation. Second, we provide more in-depth discussions on motivations, related works, and implementation (§1, §4, and §5). Third, we conduct more ablation studies to thoroughly examine each key component of the proposed method, on top of which we provide more insights (§5). Fourth, extensive experiments on two challenging datasets are performed to demonstrate the effectiveness of learning global temporal contexts (§5). Last but not least, we provide more visual results (Fig. 4) to better show the advantages of CFFM and CFFM++.

2 Related Work

2.1 Image Semantic Segmentation

Image semantic segmentation has always been a key topic in the vision community, mainly because of its wide applications in real-world scenarios. Since the pioneer work of FCN [2] which adopts fully convolution networks to make densely pixel-wise predictions, a number of segmentation methods have been proposed with different motivations or techniques [54]–[64]. For example, some works try to design effective encoder-decoder network architectures to exploit multi-level features from different network layers [2]–[5], [28]. Some works impose extra boundary supervision to improve the prediction accuracy of details [33], [65]–[69]. Some works utilize the attention mechanism to enhance the semantic representations [31], [32], [70]–[75]. Besides these talent works, we want to emphasize that most research aims at learning powerful contextual information [18]–[24], [26], [33], [76], including multi-scale [24], [26]–[28], [30], [77] and global/long-range information [18], [29], [31], [32]. The contextual information is also essential for VSS, but video contexts are different from image contexts, as discussed above.

2.2 Video Semantic Segmentation

Since the real world is dynamic rather than static, VSS is necessary for pushing semantic segmentation into more practical deployments. Previous research on VSS was limited by the available datasets [17]. Specifically, three datasets were available: Cityscapes [7], NYUDv2 [15], and CamVid [16]. They either only annotate several nonadjacent frames in a video clip or have a small scale, a low frame rate, and low resolution. Fortunately, the recent establishment of the large-scale, fully-annotated VSPW dataset [17] solves this problem.

Most of the existing VSS methods utilize the optical flow to capture temporal relations [11], [13], [14], [34], [35], [37], [38], [40], [42], [78], [79]. These methods usually adopt different smart strategies to balance the trade-off between accuracy and efficiency [78], [79]. Among them, some works aim at improving the segmentation accuracy by exploiting the temporal relations using the optical flow for feature warping [11], [13], [14] or the GAN-like architecture [80] for predictive feature learning [12]. The other works aim at improving the segmentation efficiency by using temporal consistency for feature propagation and reuse [37], [38], [40], [41], or directly reusing high-level features [37], [39], or adaptively selecting the key frame [34], or propagating segmentation results to neighbouring frames [42], or extracting features from different frames with different sub-networks [36], or considering the temporal consistency as extra training constraints [35]. Zhu *et al.* [81] utilized video prediction models to predict future frames as well as future segmentation labels, which are used as augmented data for training better image semantic segmentation models, not for VSS. Different from the above approaches, STT [82] and LMANet [83] directly model the interactions between the target and reference frame features to exploit the temporal information.

The above VSS approaches explore the local temporal relation, here denoted as *motional contexts*. However, *local temporal contexts* include two aspects: *static and motional contexts*. Those methods ignore the static contexts that are important for segmenting complicated scenes. This paper addresses this problem by proposing a new video context learning mechanism, capable of learning a unified representation of static and motional contexts. Besides, we also propose to explicitly learn *global temporal contexts* with prototype learning and attention-based feature mining.

2.3 Difference with STT

We notice that a concurrent work STT [82] also utilizes bigger searching regions for more distant video frames and self-attention for establishing connections across frames. While the two works share these similarities, there are key differences between them. *First*, two methods have different motivations. We target exploiting both static and motional contexts (local temporal contexts), while STT focuses on capturing the temporal relations among complex regions. Note that the concept of static/motional contexts is similar to the concept of simple/complex regions in STT. As a result, STT models only the motional contexts, while our method models both static and motional contexts. *Second*, the designs are different. For query selection, STT selects 50% of query locations in order to reduce the computation. However, our method splits the query features into windows and the query features in each window share the same contexts to reduce the computation. For key/value selection, STT operates in the same granularity, while our method processes the selected key/value into different granularity, which reduces the number of tokens and models the multi-scale information for static contexts. *Third*, our cross-frame feature mining can exploit multiple transformer layers to deeply mine the contextual information from the reference frames, but STT only uses one layer. The reason may be that STT only updates the query features of the selected locations and using multiple STT layers could lead to inconsistency in the query features in unselected and selected locations. Moreover, this paper also exploits global temporal contexts for further improvement.

2.4 Vision Transformer

Vision transformer, a strong competitor of convolutional neural networks (CNNs), has been widely adopted in various vision tasks [45], [48], [84]–[92], due to its powerful ability of modeling global connection within all the input tokens. Specifically, ViT [45] splits an image into patches to construct tokens and processes tokens using typical transformer layers. Swin Transformer [48] improves ViT by introducing shifted windows when computing self-attention. The effectiveness of transformers has been validated in tracking [93], [94], crowd counting [92], [95], multi-label classification [96] and so on. In the following, we specifically discuss the transformer-based segmentation methods.

To improve segmentation using transformers, some methods [52], [89], [97]–[101] have been developed. SETR [89] and Panoptic SegFormer [99] are the first transformer-based models for image and panoptic semantic segmentation, respectively. Generally, these works use transformers to generate global-context-aware features. Differently, a new trend of works such as MaskFormer [100] and Mask2Former [101] use transformer decoders to get rid of the conventional per-pixel classification for segmentation. ViT-Adapter [102] learns powerful representations from large-scale multi-modal data and allows plain ViT to achieve comparable performance to vision-specific transformers. For video understanding, [103] and [104] exploit transformers to merge temporal information and achieve promising results on the video panoptic segmentation task. Despite the success of transformers in segmentation, the use of transformer layers in VSS is non-trivial due to the large number of tokens from video frames. Here, we propose an effective and efficient way to model the

temporal contextual information for VSS. A concurrent work MRCFA [105] also works on VSS using transformers. However, MRCFA specifically focuses on refining feature affinity maps, while this paper focuses on learning local and global temporal contexts for the video.

In terms of designing vision transformers to use contextual information, Focal Transformer [85] introduces both fine-grained and coarse-grained attention in architecture design to explore local and global contexts in the image. Though our proposed methods also focus on learning contexts, there are significant differences. First, our methods focus on *video contexts*, while the Focal Transformer explores *image contexts*. Video contexts are much more complex than the image contexts. As illustrated in Fig. 1, video contexts include *local temporal* contexts which represent the contexts from neighboring frames, and *global temporal* contexts which mean the contexts from the whole video. For local temporal contexts, they can be further divided into *static* and *motional* contexts. However, the image contexts studied in Focal Transformer only refer to the information from a local region of the image or the global region (the whole image). Second, adding a new temporal dimension makes the problem of learning contexts much more challenging and significantly increases implementation difficulty. Third, our methods specifically focus on the VSS task, which are built upon a pre-trained backbone/encoder. We achieve promising performance on popular VSS datasets. Differently, the Focal Transformer proposes a new network architecture/encoder and focuses on image understanding tasks such as image classification, detection, and segmentation.

3 Local Temporal Contexts

In this section, we focus our discussion on local temporal contexts. To begin with, we introduce the technical motivation of the proposed Coarse-to-Fine Feature Mining (CFFM) for mining the local temporal contexts in §3.1. Then, we introduce the first sub-operation of Coarse-to-Fine Feature Assembling (CFFA) in §3.2. Next, we present the second sub-operation of Cross-frame Feature Mining (CFM) in §3.3. At last, we analyze the complexity in §3.4.

3.1 Technical Motivation

Before introducing our method, we discuss our technical motivation to help readers better understand the proposed technique. As discussed above, local temporal contexts include static contexts and motional contexts. The former has been well exploited in image semantic segmentation [18]–[24], [26]–[33], [77], [106], while the latter has been studied in VSS [11], [13], [14], [34], [35], [37]–[42], [78], [79]. However, there is no research touching the joint learning of both static and motional contexts which are both essential for VSS.

To address this problem, a naïve solution is to simply apply the recently popular self-attention mechanism [44]–[46] to the video sequence by viewing the feature vector at each pixel of each frame as a token. In this way, we can model global relationships by connecting each pixel with all others, so all local temporal contexts can of course be constructed. However, this naïve solution has *three obvious drawbacks*. First, a video sequence has $l+1$ times more tokens than a single image, where $l+1$ is the length of the video sequence. This would lead to $(l+1)^2$ times more computational cost than a single image because the complexity of the self-attention mechanism is $\mathcal{O}(n^2c)$, where n is the number of tokens and c is the feature dimension [44], [45], [48]. Such high complexity is unaffordable, especially for VSS which needs on-time processing as the video data stream comes in sequence. Second, such direct global modeling would be redundant. Despite that there are some motions in a video clip, the overall semantics/environment would not change suddenly and most video content is repeated. Hence, most of the (self-to-self) connections built by direct global modeling are unnecessary. Last but not least, although self-attention can technically model global relationships, a too-long sequence length would limit its performance, as demonstrated in [47]–[51], [107] where downsampling features into small scales leads to better performance than the original long sequence length.

Instead of directly modeling global relationships, we propose to model relationships only among necessary tokens for the joint learning of static and motional contexts. Our CFFM technique consists of two steps. The first step, Coarse-to-Fine Feature Assembling (CFFA), assembles the features extracted from neighbouring frames in a temporally coarse-to-fine manner based on *three observations*. First, the moving objects/stuff can only move gradually across frames in practice, and the objects/stuff cannot move from one position to another far position suddenly. Thus, the region of the possible positions of (a) moving object/stuff in a frame gradually gets larger for farther frames. In other words, for one pixel in a frame, the farther the frames, the larger the correlated regions. Second, although some content may change across frames, the overall semantics and environment would not change much, which means that most video content may only have a little temporal inconsistency. For statistical evidence, we compute the mIoU between the ground-truth masks of consecutive video frames on the VSPW val set [17], to show that the semantic masks for consecutive frames are largely overlapped and the scene changes are thus very small from a frame to its next frame. The obtained mIoU is 89.7%, proving that the objects/background move slowly from frame to frame. Third, the little temporal inconsistency of the "static" content across neighbouring frames can be easily handled by the pooling operation which is scale- and rotation-invariant, as evidenced in previous works [2], [18], [22], [29]. Inspired by the second and third observations, a varied-size region sampling through the pooling operation in neighbouring frames can convey multi-scale contextual information. Therefore, the designed CFFA can perceive multi-scale contextual information (static contexts) and motional contexts. Specifically, each pixel in the target frame corresponds to a larger receptive field and a more coarse pooling in the farther frame, as depicted in Fig. 2. Note that the length of the sampled tokens is much shorter than that in the default self-attention.

The second step of CFFM, Cross-frame Feature Mining (CFM), is designed to mine useful information from the features of neighbouring frames. This is an attention-based process. However, unlike traditional self-attention [44]–[46] whose query, key, and value come from the same input, we propose to use a *non-self attention* mechanism, where the query is from the target frame and the key and value are from neighbouring frames. Besides, we only update the query during the iterative running of non-self attention, but we keep the context tokens unchanged. This is intuitive as our goal is to mine information from neighbouring frames and the update of context tokens is thus unnecessary. Compared with self-attention which needs to process all assembled features, this non-self attention further reduces the computational cost.

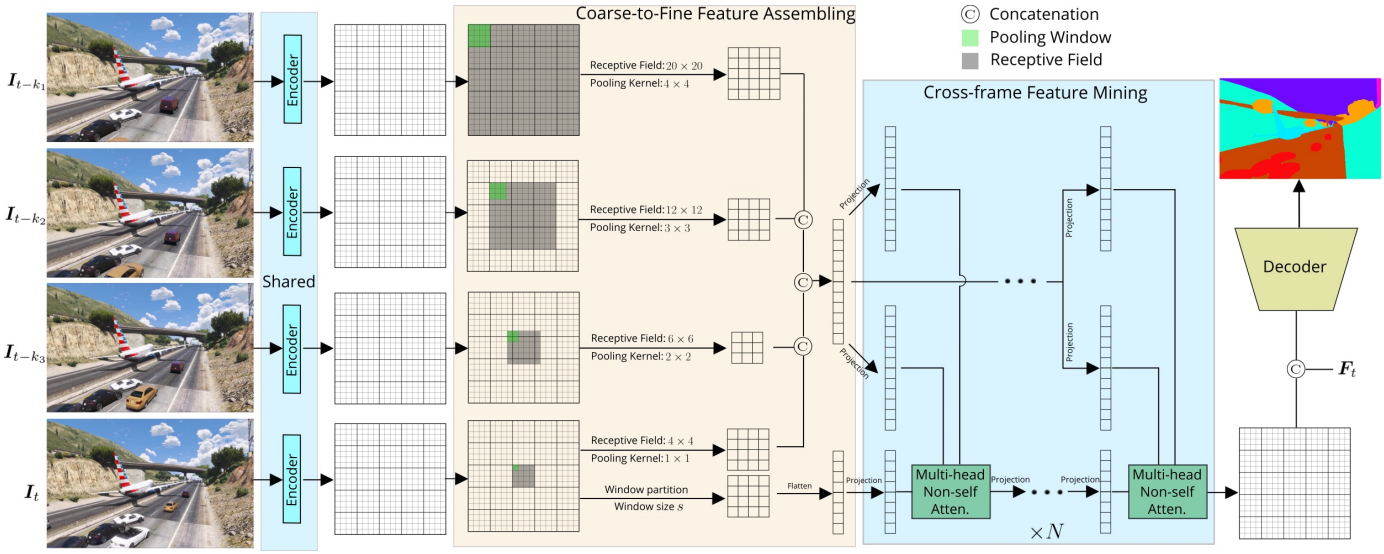


Fig. 2. **Overview of the proposed Coarse-to-Fine Feature Mining for mining local temporal contexts.** All frames are first input to an encoder to extract features, which then go through the coarse-to-fine feature assembling module (CFFA). Features for different frames are processed by different pooling strategies to generate the context tokens. The principle is that for more distant frames, a bigger receptive field and more coarse pooling are used. The shown feature size (20×20), receptive field, and pooling kernel are for a simple explanation. The context tokens from all frames are concatenated and then processed by the cross-frame feature mining (CFM) module. The context tokens are exploited to update the target features by several multi-head non-self attention layers. Finally, we use the enhanced target features to make the segmentation prediction for the target frame. *Best viewed with zooming.*

3.2 Coarse-to-Fine Feature Assembling

Without loss of generalizability, we start our discussion on training data containing nearby video frames $\{\boldsymbol{I}_{t-k_1}, \cdots, \boldsymbol{I}_{t-k_l}, \boldsymbol{I}_t\}$ with ground-truth masks of $\{\boldsymbol{S}_{t-k_1}, \cdots, \boldsymbol{S}_{t-k_l}, \boldsymbol{S}_t\}$, and we focus on segmenting $\boldsymbol{I}_t$. Specifically, $\boldsymbol{I}_t$ is the target frame and $\{\boldsymbol{I}_{t-k_1}, \cdots, \boldsymbol{I}_{t-k_l}\}$ are l previous reference frames which are $\{k_1, \cdots, k_l\}$ frames away from $\boldsymbol{I}_t$, respectively. Here, the local temporal contexts are considered since the reference frames are close to the target one. Let us denote $U = \{t - k_1, \cdots, t - k_l, t\}$ as the set of frame subscripts. We first process $\{\boldsymbol{I}_{t-k_1}, \cdots, \boldsymbol{I}_{t-k_l}, \boldsymbol{I}_t\}$ using an encoder to extract informative features $\{\boldsymbol{F}_{t-k_1}, \cdots, \boldsymbol{F}_{t-k_l}, \boldsymbol{F}_t\}$, each of which has the size of $\mathbb{R}^{h \times w \times c}$ (h, w, and c represent height, width, and feature dimension, respectively). We aim to exploit the features from the nearby frames to generate better features for segmenting $\boldsymbol{I}_t$ as valuable local temporal contexts exist in previous frames.

To efficiently establish long-range interactions between the reference frame features ($\{\boldsymbol{F}_{t-k_1}, \cdots, \boldsymbol{F}_{t-k_l}\}$) and the target frame features $\boldsymbol{F}_t$, we propose the coarse-to-fine feature assembling module, as shown in Fig. 2. Inspired by previous works [48], [85], [107], we split the target frame features $\boldsymbol{F}_t$ into windows and each window attends to a shared context token. The reason behind this is that attending each location in $\boldsymbol{F}_t$ to a specific context token requires huge computation and memory costs. When using window size of $s \times s$, $\boldsymbol{F}_t$ is partitioned into $\frac{h}{s} \times \frac{w}{s}$ windows. We obtain the new feature map $\boldsymbol{F}'_t$ as follows:

$$\begin{aligned} \boldsymbol{F}_t \in \mathbb{R}^{h \times w \times c} &\rightarrow \boldsymbol{F}'_t \in \mathbb{R}^{(\frac{h}{s} \times s) \times (\frac{w}{s} \times s) \times c} \\ &\rightarrow \boldsymbol{F}'_t \in \mathbb{R}^{\frac{h}{s} \times \frac{w}{s} \times s \times s \times c}. \end{aligned} \tag{1}$$

Then, we generate context tokens from different frames. The main idea is to see a bigger receptive field and use a more coarse pooling if the frame is more distant from the target, which is why we call this step coarse-to-fine feature assembling. The motivation behind this is described in §3.1. Formally, we define two sets of parameters: the receptive fields $r = \{r_{t-k_1}, \cdots, r_{t-k_l}, r_t\}$ and the pooling kernel/window sizes $p = \{p_{t-k_1}, \cdots, p_{t-k_l}, p_t\}$, when generating corresponding context tokens. For $t - k_1 < t - k_2 < \cdots < t - k_l < t$, we have $r_{t-k_1} \geq r_{t-k_2} \geq \cdots \geq r_{t-k_l} \geq r_t$ and $p_{t-k_1} \geq p_{t-k_2} \geq \cdots \geq p_{t-k_l} \geq p_t$. With this definition, we partition $\{\boldsymbol{F}_{t-k_1}, \cdots, \boldsymbol{F}_{t-k_l}, \boldsymbol{F}_t\}$ using pooling windows $p = \{p_{t-k_1}, \cdots, p_{t-k_l}, p_t\}$ to pool the features, respectively. The result is processed by a fully connected layer (FC) for dimension reduction. This is formulated as

$$\begin{aligned} \boldsymbol{F}_j \in \mathbb{R}^{h \times w \times c} &\rightarrow \boldsymbol{E}_j \in \mathbb{R}^{\frac{h}{p_j} \times \frac{w}{p_j} \times (p_j \times p_j \times c)} \\ &\xrightarrow{\text{FC}} \boldsymbol{E}_j \in \mathbb{R}^{\frac{h}{p_j} \times \frac{w}{p_j} \times c}, \end{aligned} \tag{2}$$

where $j \in U$. In Fig. 2, we have $r = \{20, 12, 6, 4\}$ and $p = \{4, 3, 2, 1\}$ for all frames (3 reference and 1 target frames).

For each window partition $\boldsymbol{F}'_t[i] \in \mathbb{R}^{s \times s \times c}$ ($i \in \{1, 2, \cdots, \frac{hw}{s^2}\}$) in the target features, we extract $\frac{r_j}{p_j} \times \frac{r_j}{p_j}$ elements from $\boldsymbol{E}_j$ around the area where the window lies in. This can be easily implemented using the *unfold* function in PyTorch [108]. Let $\boldsymbol{c}_{i,j}$ denote the obtained context tokens from j-th frame and for i-th window partition in the target features. We concatenate $\boldsymbol{c}_{i,j}$ into $\boldsymbol{c}_i$ as follows,

$$\boldsymbol{c}_i = \text{Concat}[\boldsymbol{c}_{i,j}], \tag{3}$$

where $j \in U$, $\boldsymbol{c}_i \in \mathbb{R}^{m \times c}$ and $m = \sum_{j \in U} \frac{r_j^2}{p_j^2}$. The context tokens from the target frame are obtained by using the parameter set (r_t, p_t) to process the target features. In practice, we additionally use another parameter set (r'_t, p'_t) to generate more contexts from the target since the target features are more important. For simplicity, we focus our discussion by omitting (r'_t, p'_t) and using only (r_t, p_t) for the target.

To sum up, $\boldsymbol{c}_i$ contains the context information from all frames, which is used to refine the target frame features. As

discussed in §3.1, on one hand, $\boldsymbol{c}_i$ covers the tokens at possible positions where moving objects/stuff would appear, so it can be used for learning motional contexts. On the other hand, $\boldsymbol{c}_i$ is a multi-scale sampling of neighbouring frames with the temporal inconsistency solved by the pooling operation, so it can be used for learning static contexts.

3.3 Cross-frame Feature Mining

After that we obtain the context token $\boldsymbol{c}_i$ for each window partition in the target features, we propose a non-self attention mechanism to mine useful information from neighboring frames. Unlike the traditional self-attention mechanism that computes the query, key, and value from the same input, our non-self attention mechanism utilizes different inputs to calculate the query, key, and value. Since $\boldsymbol{F}_t'$ is the input to the first layer of our CFM module, we re-write it as $\boldsymbol{F}_t^0 = \boldsymbol{F}_t'$. For the i-th window partition in $\boldsymbol{F}_t^0$, the query Q_i, key K_i, and value V_i are computed using three fully connected layers as follows:

$$Q_i = \mathrm{FC}(\boldsymbol{F}_t^0[i]), \quad K_i = \mathrm{FC}(\boldsymbol{c}_i), \quad V_i = \mathrm{FC}(\boldsymbol{c}_i), \tag{4}$$

where $\mathrm{FC}(\cdot)$ represents an FC layer. Next, we use non-self attention to update the target frame features, given by

$$\boldsymbol{F}_t^1[i] = \mathrm{Softmax}(\frac{Q_i K_i^T}{\sqrt{c}} + B)V_i + \boldsymbol{F}_t^0[i], \tag{5}$$

where B represents the position bias, following [48]. Note that we omit the formulation of the multi-head attention [44], [45] for simplicity. Eq. (4) and Eq. (5) are repeated for N steps, and we finally obtain the enhanced feature $\boldsymbol{F}_t^N \in \mathbb{R}^{\frac{h}{s}\times\frac{w}{s}\times s\times s\times c}$ for the target video frame. Local temporal contexts, *i.e.*, static and motional contexts, from neighbouring video frames are continuously exploited to learn better representative features for segmenting the target frame. Note that in this process, we do not update the context tokens $\boldsymbol{c}_i$ for simplicity/elegance and reducing computation. Since this step is to mine useful information from the reference frames, it is also unnecessary to update $\boldsymbol{c}_i$. This is the advantage of non-self attention.

To generate segmentation predictions, we reshape $\boldsymbol{F}_t^N$ into $\mathbb{R}^{h\times w\times c}$ and concatenate $\boldsymbol{F}_t^N$ with $\boldsymbol{F}_t$. Then, a simple MLP projects the features to segmentation logits $\boldsymbol{R}_t$. The common cross-entropy loss (CE) is computed between $\boldsymbol{R}_t$ and ground-truth mask $\boldsymbol{S}_t$. Auxiliary losses on original features are also computed. During inference, our method does not need to extract features for all $l+1$ frames when processing $\boldsymbol{I}_t$. Instead, the features of the reference frames, which are the frames before the target frame, have already been extracted in previous steps. Only the target frame is passed to the encoder to generate $\boldsymbol{F}_t$, and then features $\{\boldsymbol{F}_{t-k_1}, \cdots, \boldsymbol{F}_{t-k_l}, \boldsymbol{F}_t\}$ for all frames are passed to CFFM for representation enhancement.

3.4 Complexity Analysis

Here, we formally analyze the complexity of the proposed CFFM and the recent popular self-attention mechanism [44]–[46] when processing video clip features $\{\boldsymbol{F}_{t-k_1}, \cdots, \boldsymbol{F}_{t-k_l}, \boldsymbol{F}_t\}$. The coarse-to-fine feature assembling (Eq. (2)) has the complexity of $\mathcal{O}((l+1)hwc)$, which is irrespective of p. The cross-frame feature mining has two parts: Eq. (4) has the complexity of $\mathcal{O}(hwc^2) + \mathcal{O}(mc^2)$, and Eq. (5) is with the complexity of $\mathcal{O}(hwmc)$. As mentioned early, $m = \sum_{j\in U}\frac{r_j^2}{p_j^2}$. To sum over, the complexity of our method is given by

$$\begin{aligned}\mathcal{O}(\mathrm{CFFM}) &= \mathcal{O}(hwmc) + \mathcal{O}(hwc^2) + \mathcal{O}(mc^2) \\ &\quad + \mathcal{O}((l+1)hwc) \\ &= \mathcal{O}(hwmc) + \mathcal{O}(hwc^2),\end{aligned} \tag{6}$$

where the derivation is conducted by removing less significant terms. For the self-attention mechanism [44]–[46], the complexity is $\mathcal{O}((l+1)^2h^2w^2c) + \mathcal{O}((l+1)hwc^2)$. Since $m \ll (l+1)^2hw$, the complexity of the proposed approach is much less than the self-attention mechanism. Take the example in Fig. 2, $m = 66$ while $(l+1)^2hw = 6400$.

4 Global Temporal Contexts

In this section, we focus our discussion on the global temporal contexts. We start by explaining the process of extracting global temporal contextual information (prototypes). Then, we discuss how to exploit the generated contextual prototypes to refine the features of the target frame.

4.1 Global Temporal Contextual Prototypes

In the last section (§3), we discuss how to learn local temporal contexts among nearby video frames $\{\boldsymbol{I}_{t-k_1}, \cdots, \boldsymbol{I}_{t-k_l}, \boldsymbol{I}_t\}$. Here, we propose to learn global temporal contexts to make the model have a much larger temporal view. To start with, we represent the corresponding whole video as $\boldsymbol{V} = \{\boldsymbol{I}_1, \cdots, \boldsymbol{I}_t, \cdots, \boldsymbol{I}_T\}$, containing a total of T frames. Similar to §3.2, we aim to segment the target frame $\boldsymbol{I}_t$ without losing generalizability. In the following, we explain the process of extracting the global temporal contextual information.

Our technique is built on the CFFM introduced in §3. Once CFFM is trained, the encoder has the ability to generate informative features for each frame of the video. We first use the trained encoder to extract features for a subset of frames in the video $\boldsymbol{V}$. Specifically, the subset of frames are uniformly sampled from $\boldsymbol{V}$ by a fixed step d, which are denoted as $\bar{\boldsymbol{V}} = \{\boldsymbol{I}_1, \boldsymbol{I}_{1+d}, \cdots, \boldsymbol{I}_{1+(f-1)*d}\}$. Here, a total of f frames are sampled and $d = \left\lfloor \frac{T}{f} \right\rfloor$. We conduct the sampling for three reasons: 1) the original video $\boldsymbol{V}$ has many frames (average $T = 71$ for VSPW [17]), and it is unaffordable to explicitly extract global temporal contexts from all T frames; 2) the contents in nearby frames have already been modeled by CFFM through static and motional contexts; 3) most contents in nearby frames are redundant and the sampled $\bar{\boldsymbol{V}}$ contains enough contexts from a global temporal view. The corresponding extracted features for the sampled frames are $\{\boldsymbol{F}_1, \boldsymbol{F}_{1+d}, \cdots, \boldsymbol{F}_{1+(f-1)*d}\}$.

Next, we tokenize the extracted features and treat the feature vector at each pixel as a token, resulting in tokens $\boldsymbol{o} \in \mathbb{R}^{N_o\times c}$, where $N_o = fhw$ is the total number of tokens. Those tokens contain the global temporal contexts for the whole video. However, it is impractical to exploit $\boldsymbol{o}$ due to the large number of tokens. Inspired by prototype learning [109], [110], we exploit unsupervised clustering to extract typical contextual prototypes from $\boldsymbol{o}$. This largely reduces the number of tokens for the following processing, thus saving computational resources. The extracted prototypes still contain the necessary and relevant contexts for the whole video, while having a much smaller size and more condensed information. In our experiments, we use k-means to generate contextual prototypes $\boldsymbol{p} \in \mathbb{R}^{N_p\times c}$ from $\boldsymbol{o} \in \mathbb{R}^{N_o\times c}$,

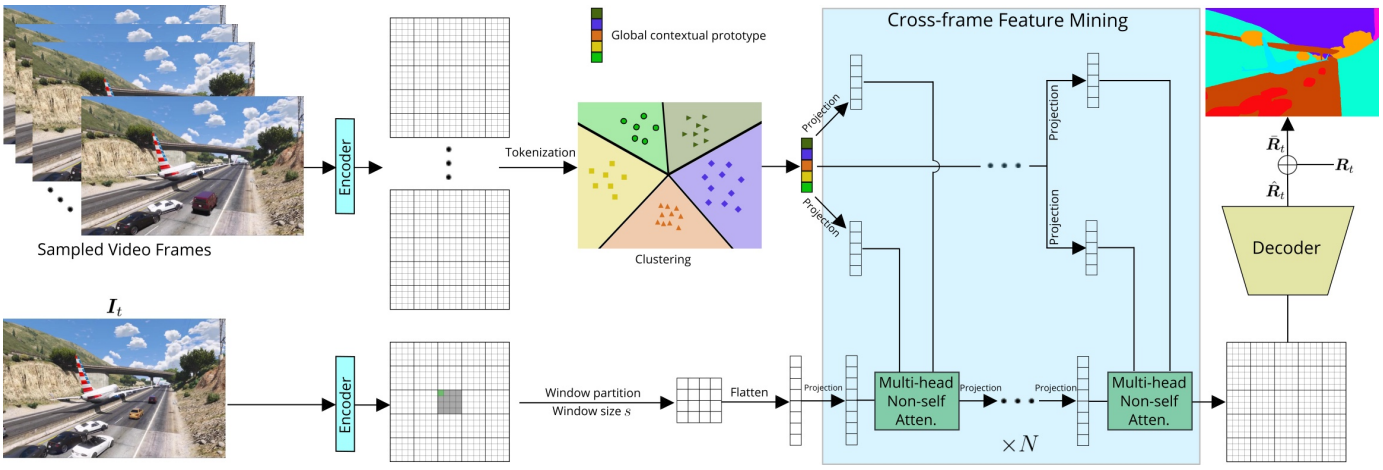


Fig. 3. **Overview of the proposed CFFM++ for additionally mining global temporal contexts.** Due to the large number of frames in the video, we uniformly sample frames by a fixed step. The sampled video frames go through the encoder trained by CFFM and corresponding features are generated. After tokenizing the feature maps, we conduct unsupervised clustering (k-means) to reduce the tokens' number and learn global contextual prototypes. The obtained prototypes and the target frame features are passed to CFM, which enables the refinement of the target frame using global temporal contexts. The final predictions of CFFM++ are given by the weighted summation of the segmentation logits from learning local (CFFM) and global temporal contexts.

where $N_p \ll N_o$. In our experiments, N_p is set to 100 unless otherwise specified.

Due to the selection of video frames across the whole video and the use of GPU-based k-means clustering, the process of generating global temporal contextual prototypes is fast and does not significantly decrease the speed, which will be shown in our experiments (§5.2).

4.2 Global Temporal Context Mining

After obtaining global temporal contextual prototypes $\boldsymbol{p}$, we again exploit the CFM module to mine the global temporal contexts for refining the features of the target frame. $\boldsymbol{F}_t'$ is input to the first layer of the CFM module and we re-write it as $\boldsymbol{G}_t^0 = \boldsymbol{F}_t'$. Specifically, for the i-th window partition in $\boldsymbol{G}_t^0$ ($\boldsymbol{F}_t'$), the query Q_i, key K_i, and value V_i are computed using three fully connected layers as follows:

$$Q_i = \mathrm{FC}(\boldsymbol{G}_t^0[i]), \quad K_i = \mathrm{FC}(\boldsymbol{p}), \quad V_i = \mathrm{FC}(\boldsymbol{p}). \tag{7}$$

Here, the contextual prototypes $\boldsymbol{p}$ contain the contexts from the whole video and are shared for all the patches of the target frame in the video. Next, we use non-self attention to update the target frame features, as follows:

$$\boldsymbol{G}_t^1[i] = \mathrm{Softmax}(\frac{Q_i K_i^T}{\sqrt{c}})V_i + \boldsymbol{G}_t^0[i], \tag{8}$$

Similar to CFFM (§3.3), Eq. (7) and Eq. (8) are repeated for N_g times. After refined by the global temporal contextual prototypes, the final feature for the target frame is given by $\boldsymbol{G}_t^{N_g}$, which is reshaped into $\mathbb{R}^{h\times w\times c}$.

To generate segmentation predictions, a simple MLP is used to project $\boldsymbol{G}_t^{N_g}$ into segmentation logits $\hat{\boldsymbol{R}}_t$. During training, the cross entropy loss (CE) is computed between $\hat{\boldsymbol{R}}_t$ and $\boldsymbol{S}_t$. During inference, we combine the logits learned from local and global temporal contexts in a weighted manner, *i.e.*, $\bar{\boldsymbol{R}}_t = \lambda\hat{\boldsymbol{R}}_t + \boldsymbol{R}_t$. In our experiment, we set λ to be 0.5. The method using predictions given by $\bar{\boldsymbol{R}}_t$ is denoted as CFFM++, which is the extended version of CFFM by additionally exploiting global temporal contexts.

5 Experiments

5.1 Experimental Setup

Implementation details for CFFM. We implement our approach based on the `mmsegmentation` [112] codebase and conduct all experiments on 4 NVIDIA RTX 6000 GPUs (24G memory). The backbones are the same as SegFormer [52], which are all pretrained on ImageNet [113]. For other parts of our model, we adopt random initialization. Our model uses 3 reference frames unless otherwise specified, and we have $\{k_1, k_2, k_3\} = \{9, 6, 3\}$, following [17]. We found that this selection of reference frames is enough to model *local temporal contexts* and achieve impressive performance. For the receptive field, pooling kernel, and window size, we set $r = \{49, 20, 6, 7\}$, $p = \{7, 4, 2, 1\}$, and $s = 7$. For the target frame, we additionally have $r_t' = 35$ and $p_t' = 5$. During training, we adopt augmentations including random resizing, flipping, cropping, and photometric distortion. We use the crop size of 480×480 for the VSPW dataset [17] and 512×1024 for Cityscapes [7]. For optimizing parameters, we use the AdamW and "poly" learning rate schedule with an initial learning rate of $6e$-5. The network is trained for 160k iterations, following SegFormer [52]. During testing, we conduct single-scale testing and resize all images on VSPW to the size of 480×853 and 512×1024 for Cityscapes. Note that for efficiency and simplicity, the predicted mask is obtained by feeding the whole image to the network, rather than using the sliding window as in [89]. We do *not* use any post-processing such as CRF [114].

Impelmentation details for CFFM++. Our CFFM++ is built on CFFM. Once CFFM is trained, the corresponding encoder has the ability to extract informative features from video frames. Hence, we use the trained encoder from CFFM as the feature extractor for generating the global temporal contextual prototypes (§4.1). When generating prototypes, we set the number of sampled video frames (f) as 10 for all videos. The number of prototypes $N_p = 100$. When mining the global temporal contexts, we set N_g as 1 for small models (MiT-B0 and MiT-B1) and 2 for large models. During the training of CFFM++, we freeze the encoder and CFFM parameters, while only updating the multi-head non-

TABLE 1
Comparison with state-of-the-art methods on the VSPW [17] validation set. Our models outperform the compared methods, with better balance in terms of model size, accuracy, latency, and speed. Both FPS and MACs are computed with the input size of 480×853.

Methods	Backbone	Params (M) ↓	mIoU ↑	Weighted IoU ↑	mVC_8 ↑	mVC_{16} ↑	MACs (G) ↓	FPS (f/s) ↑
SegFormer [52]	MiT-B0	3.8	32.9	56.8	82.7	77.3	14.6	73.4
SegFormer [52]	MiT-B1	13.8	36.5	58.8	84.7	79.9	33.0	58.7
CFFM (Ours)	MiT-B0	4.7	35.4	58.5	87.7	82.9	25.6	43.1
CFFM (Ours)	MiT-B1	15.5	38.5	60.0	88.6	84.1	55.4	29.8
CFFM++ (Ours)	MiT-B0	5.7	35.9	58.9	88.4	83.8	34.3	40.4
CFFM++ (Ours)	MiT-B1	16.5	**39.9**	**60.7**	**89.1**	**84.9**	64.2	27.6
DeepLabv3+ [28]	ResNet-101	62.7	34.7	58.8	83.2	78.2	-	-
UperNet [111]	ResNet-101	83.2	36.5	58.6	82.6	76.1	-	-
PSPNet [29]	ResNet-101	70.5	36.5	58.1	84.2	79.6	-	13.9
OCRNet [21]	ResNet-101	58.1	36.7	59.2	84.0	79.0	-	14.3
ETC [35]	PSPNet	89.4	36.6	58.3	84.1	79.2	-	-
NetWarp [111]	PSPNet	89.4	37.0	57.9	84.4	79.4	-	-
ETC [35]	OCRNet	58.1	37.5	59.1	84.1	79.1	-	-
NetWarp [111]	OCRNet	58.1	37.5	58.9	84.0	79.0	-	-
$\text{TCB}_{\text{st-ppm}}$ [17]	ResNet-101	70.5	37.5	58.6	87.0	82.1	-	10.0
$\text{TCB}_{\text{st-ocr}}$ [17]	ResNet-101	58.1	37.4	59.3	86.9	82.0	-	5.5
$\text{TCB}_{\text{st-ocr-mem}}$ [17]	ResNet-101	58.1	37.8	59.5	87.9	84.0	-	5.5
SegFormer [52]	MiT-B2	24.8	43.9	63.7	86.0	81.2	57.2	39.2
SegFormer [52]	MiT-B5	82.1	48.2	65.1	87.8	83.7	187.0	17.2
CFFM (Ours)	MiT-B2	26.5	44.9	64.9	89.8	85.8	79.6	23.8
CFFM (Ours)	MiT-B5	85.5	49.3	65.8	**90.8**	87.1	232.2	11.3
CFFM++ (Ours)	MiT-B2	28.5	45.5	64.7	90.2	86.4	96.9	21.5
CFFM++ (Ours)	MiT-B5	87.5	**50.1**	**66.5**	**90.8**	**87.4**	249.5	10.4

TABLE 2
Comparison with state-of-the-art methods on the VSPW [17] test set. Our model outperforms the compared methods. * means the test results are from [17].

Methods	Backbone	Params (M)	mIoU	mVC_8	mVC_{16}
SegFormer [52]	MiT-B0	3.8	30.9	81.6	75.7
SegFormer [52]	MiT-B1	13.8	33.5	82.6	76.9
CFFM (Ours)	MiT-B0	4.7	31.8	86.3	80.9
CFFM (Ours)	MiT-B1	15.5	35.1	87.2	82.2
CFFM++ (Ours)	MiT-B0	5.7	32.8	87.4	82.4
CFFM++ (Ours)	MiT-B1	16.5	**36.0**	**87.9**	**83.2**
DeepLabv3+* [28]	ResNet-101	62.7	32.2	81.0	75.0
UperNet* [111]	ResNet-101	83.2	33.5	79.3	73.3
PSPNet* [29]	ResNet-101	70.5	33.8	83.4	78.3
OCRNet* [21]	ResNet-101	58.1	34.0	82.9	77.4
ETC* [35]	PSPNet	89.4	33.8	82.8	77.1
NetWarp* [111]	PSPNet	89.4	33.7	82.6	77.1
ETC* [35]	OCRNet	58.1	34.6	83.1	78.0
NetWarp* [111]	OCRNet	58.1	35.0	83.2	77.2
$\text{TCB}_{\text{st-ppm}}$* [17]	ResNet-101	70.5	34.6	85.2	80.2
$\text{TCB}_{\text{st-ocr}}$* [17]	ResNet-101	58.1	35.1	85.1	80.1
$\text{TCB}_{\text{st-ocr-mem}}$* [17]	ResNet-101	58.1	35.6	86.2	81.9
SegFormer [52]	MiT-B2	24.8	40.0	84.9	79.8
CFFM (Ours)	MiT-B2	26.5	41.0	88.4	83.6
CFFM++ (Ours)	MiT-B2	28.5	**42.0**	**88.9**	**84.7**

self attention modules since CFFM is already well-optimized. This also reduces the training iterations needed for fine-tuning the newly added non-self attention modules. For this fine-tuning, we only use 40k iterations. The learning rate is set as 2e-4. Other settings are kept the same as CFFM.

Datasets. Our experiments are mainly conducted on the VSPW dataset [17], which is the largest VSS benchmark. Its training, validation, and test sets have 2,806 clips (198,244 frames), 343 clips (24,502 frames), and 387 clips (28,887 frames), respectively. It contains diverse scenarios including both indoor and outdoor scenes, annotated for 124 categories. More importantly, VSPW has dense annotations with a high frame rate of 15fps, making itself the best benchmark for VSS till now. In contrast, previous datasets used for VSS only have very sparse annotation, *i.e.*, only one frame out of many consecutive frames is annotated. Both training and validation sets of VSPW are publicly available while the test set is not open. However, the test performance can be obtained from the VSPW2021 challenge server. On the server, the test is split into the development part and the final part, and only evaluation on the final part is available. We obtain the performance on the test set through the server. In addition to VSPW, we also evaluate the proposed method on the Cityscapes dataset [7], which annotates one frame out of every 30 frames.

Evaluation metrics. Following previous works [2], we use mean IoU (mIoU), and weigheted IoU to evaluate the segmentation performance. In addition, we also adopt video consistency (VC) [17] to evaluate the smoothness of the predicted segmentation maps in the temporal domain. Formally, for a video clips $\{\boldsymbol{I}_t\}_{t=1}^{T}$ with ground-truth segmentation masks $\{\boldsymbol{S}_t\}_{t=1}^{T}$ and predicted masks $\{\boldsymbol{S}'_t\}_{t=1}^{T}$, VC_n is computed as follows,

$$\text{VC}_n = \frac{1}{T-n+1} \sum_{i=1}^{T-n+1} \frac{(\cap_i^{i+n-1} \boldsymbol{S}_i) \cap (\cap_i^{i+n-1} \boldsymbol{S}'_i)}{\cap_i^{i+n-1} \boldsymbol{S}_i}, \quad (9)$$

where $T \geq n$. After computing VC_n for every video, we obtain the mean of VC_n for all videos as mVC_n. The purpose of this metric is to evaluate the level of consistency in the predicted masks among those common areas (pixels' semantic labels do not change) across long-range frames. For more details, please refer to [17]. Note that, to compute the VC metric, the ground-truth masks for all frames are needed.

The details of computing FPS are as follows. The FPS is measured in mini-batches with the batch size set to 2. We keep note of the computation time $\mathcal{T}$ for processing $\mathcal{K}$ mini-batches. The FPS can be calculated by $2\mathcal{K}/\mathcal{T}$. We set the batch size to 2 because this leads to high usage (>95%) of GPU, which is common in this community. We computed the FPS for all methods in the same way for fair comparisons.

TABLE 3
Comparison with recent efficient VSS methods on the Cityscapes [7] dataset. Our methods are superior to the compared methods.

Methods	Backbone	Params (M)	mIoU	FPS (f/s)
FCN [2]	MobileNetV2	9.8	61.5	14.2
CC [37]	VGG-16	-	67.7	16.5
DFF [40]	ResNet-101	-	68.7	9.7
GRFP [14]	ResNet-101	-	69.4	3.2
PSPNet [29]	MobileNetV2	13.7	70.2	11.2
DVSN [34]	ResNet-101	-	70.3	19.8
Accel [38]	ResNet-101	-	72.1	3.6
ETC [35]	ResNet-18	13.2	71.1	9.5
SegFormer [52]	MiT-B0	3.7	71.9	58.5
CFFM (Ours)	MiT-B0	4.6	74.0	34.2
CFFM++ (Ours)	MiT-B0	5.1	74.3	28.8
SegFormer [52]	MiT-B1	13.8	74.1	46.8
CFFM (Ours)	MiT-B1	15.4	75.1	23.6
CFFM++ (Ours)	MiT-B1	15.9	**75.7**	20.4

TABLE 4
Ablation study on the number of attention layers in CFM.

Methods	N	mIoU	mVC_8	mVC_{16}	Params (M)
MiT-B0					
SegFormer [52]	-	32.9	82.7	77.3	3.8
CFFM (Ours)	1	35.4	**87.7**	82.9	4.7
	2	**35.7**	**87.7**	**83.0**	5.5
MiT-B1					
SegFormer [52]	-	36.5	84.7	79.9	13.8
CFFM (Ours)	1	37.8	88.3	83.6	14.6
	2	38.5	**88.6**	**84.1**	15.5
	3	38.7	**88.6**	**84.1**	16.3
	4	**38.8**	88.5	83.9	17.2

TABLE 5
Ablation study on the selection of the reference frames. We use MiT-B1 as the backbone.

Methods	k1	k2	k3	mIoU	mVC_8	mVC_{16}
SegFormer	-	-	-	36.5	84.7	79.9
CFFM (Ours)	-	-	3	37.4	87.4	82.4
	-	-	6	37.7	88.0	83.3
	-	-	9	37.9	88.4	83.9
	3	2	1	37.7	88.3	83.6
	9	6	3	**38.5**	**88.6**	**84.1**

5.2 Comparison with State-of-the-art Methods

Results on CFFM. We compare the proposed method with state-of-the-art VSS methods on VSPW [17] in Tab. 1. The results are analyzed from different aspects. For small models (number of parameters less than 20M), CFFM outperforms corresponding baselines with a clear margin, while introducing limited model complexity. For example, using the backbone MiT-B0, CFFM has 2.5% mIoU gain over the strong baseline of SegFormer [52], with the cost of increasing the parameters from 13.8M to 15.5M, increasing MACs from 33.0G to 55.4G, and reducing the FPS (frames per second) from 73.4 (f/s) to 43.1 (f/s). Our method also provides much more consistent predictions for the videos, outperforming the baseline with 5.0% and 5.6% in terms of mVC_8 and mVC_{16}, respectively. Note that both metrics mVC_8 and mVC_{16} provide an evaluation of visual consistency within predicted masks for videos, as verified in [17].

For large models (number of parameters $>$ 20M), CFFM achieves state-of-the-art performance in this challenging dataset and also generates visually consistent results. Specifically, our model using MiT-B2 has 26.5M parameters (slightly larger than SegFormer [52]) and achieves 44.9% mIoU at the FPS of 23.8 (f/s), using 79.6G MACs. Our large model (based on MiT-B5) achieves mIoU of 49.3% and performs best in terms of visual consistency, with mVC_8 and mVC_{16} of 90.8% and 87.1%, respectively. To summarize, for all backbones (MiT-B0, MiT-B1, MiT-B2, and MiT-B5), CFFM clearly outperforms the corresponding baseline, showing that the proposed modules are stable and provide consistent performance improvement. The results validate the effectiveness of the proposed coarse-to-fine feature assembling (CFFA) and cross-frame feature mining (CFM) in mining relevant information (*local temporal contexts*) from nearby frames.

We also obtain results on the test set of the VSPW dataset from the VSPW2021 challenge server, which is shown in Tab. 2. We can observe that the proposed CFFM surpasses the considered approaches. For example, upon MiT-B1, CFFM is clearly better than the baseline (SegFormer), with an mIoU gain of 1.6%. The experimental results on the Cityscapes [7] dataset are shown in Tab. 3. Our method is compared with recent efficient segmentation methods. Only using 4.6M parameters, CFFM obtains 74.0% mIoU with an FPS of 34.2 (f/s), achieving an excellent balance on model size, accuracy, and speed. When using a deeper backbone, we achieve 75.1% mIoU with an FPS of 23.6 (f/s). Note that this dataset has sparse annotations, the excellent performance demonstrates that our method works well for both fully supervised and semi-supervised settings.

Results on CFFM++. CFFM++, the extension of CFFM by additionally exploring *global temporal contexts*, achieves consistent improvements over CFFM on the VSPW dataset under all studied backbones while introducing limited computation resources. For example, using MiT-B1, CFFM++ obtains an mIoU gain of 1.4% over CFFM and generates more visually consistent predictions with a gain of 0.8% in the metric mVC_{16}. We also compute the average improvement of CFFM++ over CFFM across various backbones. On average, CFFM++ noticeably outperforms CFFM by 0.9 and 1.0 point on the VSPW val and test sets, respectively. The improvements are valid for all datasets and backbones, showing the effectiveness of mining global temporal contexts.

We also observe that CFFM++ is efficient in terms of model size and computation speed. For example, with MiT-B1, CFFM++ only increases the number of parameters from 15.5M to 16.5M and reduces the FPS from 29.8 (f/s) to 27.6 (f/s), compared with CFFM. The reasons why the proposed CFFM++ does not noticeably increase the latency are in three aspects. First, the number of global contextual prototypes is very small, *i.e.*, 100 in our main experiments. Second, in CFM, only 2 non-self attention layers are used. Third, the decoder (Fig. 2 and Fig. 3) is very small, comprising a single convolutional layer mapping to class logits, following SegFormer [52]. It means that the additional latency caused by mining global temporal contexts (CFFM++) is very insignificant, compared to the computation of CFFM. Therefore, CFFM++ is only a little slower than CFFM.

Qualitative Results. The qualitative results are shown in Fig. 4. For the given examples, CFFM resolves the inconsistency existing in the predictions of the baseline, as it exploits the *local temporal contexts* within nearby frames. Build upon CFFM, CFFM++ further improves the per-frame prediction accuracy and temporal consistency by utilizing the *global temporal contexts* within a long range of video frames.

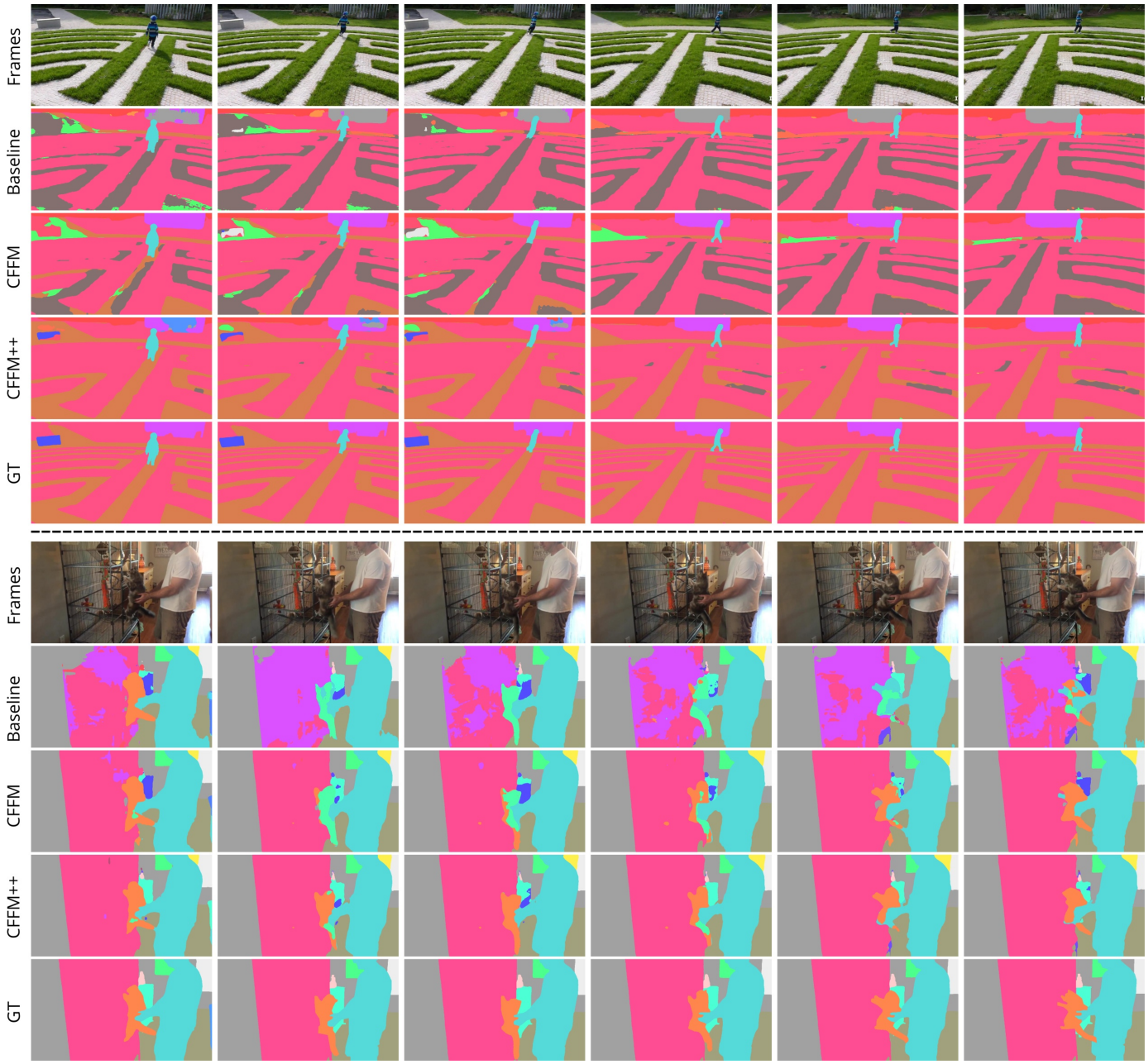


Fig. 4. **Qualitative results for two video clips.** We compare the proposed methods with the baseline (SegFormer [52]) visually. From *top* to *down*: the input video frames, the predictions of SegFormer [52], CFFM predictions, CFFM++ predictions and the ground truth (GT). It shows that CFFM produces more accurate and consistent results, compared to the strong baseline. Furthermore, by using global temporal contexts, CFFM++ further improves over CFFM. *Best viewed in color.*

5.3 Ablation Study

All ablation studies are conducted on the large-scale VSPW [17] dataset and follow the same training strategies as described above, for a fair comparison.

Influence of the number of attention layers. Tab. 4 shows the performance of CFFM with respect to the number of non-self attention layers in the CFM module. For two backbones of MiT-B0 [52] and MiT-B1 [52], CFFM clearly outperforms the corresponding baseline (SegFormer) when using only a single attention layer and introducing a small number of additional parameters. It demonstrates the effectiveness of the proposed CFFA module and the non-self attention layer. The former efficiently extracts the *local temporal contexts* from the nearby frames and the latter effectively mines the contextual information to refine target frame features. In addition, we observe there is a trade-off between performance and the model complexity (number of parameters) on the MiT-B1 backbone. When using more attention layers within the CFM module, better mIoU is obtained while the model size linearly increases. For our method (CFFM) on MiT-B1, we choose $N = 2$ since a better trade-off is observed.

Impact of selection of reference frames. We study the impact of the selection of reference frames for learning *local temporal contexts* in Tab. 5. We start by using a single reference frame. There seems to be a trend that when increasing the distance between the reference frame and the target frame, better performance is obtained. The reason for this is that the more faraway reference

frame may contain richer and more different contexts which complements the contexts of the target frame. This also suggests that extracting global temporal contexts from the whole video is useful. When using more reference frames ($k_1 = 9, k_2 = 6, k_3 = 3$), the best performance with mIoU of 38.5% is achieved. It is worth noting that CFFM using reference frames combination of $k_1 = 3, k_2 = 2$, and $k_3 = 1$, only achieves segmentation mIoU of 37.7% and performs similarly as the cases when using a single reference frame. It is possibly due to the fact that the very close reference frames do not give much new information for segmenting the target frame, as also shown in [17].

Impact of CFFA and CFM. Starting from SegFormer [52], we only add CFFA to extract contextual tokens. To mine the generated contexts, we use an MLP to process them, which are finally merged with the target features. For this variant of only using CFFA, we obtain a mIoU of 37.6%, outperforming the baseline with a mIoU of 36.5% by 1.1% gain. Then, we add both CFFA and CFM on top of the baseline, which is our final model (CFFM) for learning *local temporal contexts*. The segmentation performance (mIoU) for CFFM is 38.5%. These facts verify that both CFFA and CFM modules are valuable and essential to the proposed CFFM mechanism for learning local temporal contexts.

Impact on the Receptive Fields. To investigate the impact of receptive fields, we conduct ablation study on different $r = \{r_{t-k_1}, r_{t-k_2}, r_{t-k_3}, r\}$. As mentioned previously, we use three reference frames, with $k_1 = 9$, $k_2 = 6$, and $k_3 = 3$, following [17]. Note that larger receptive fields mean that the larger region/context is used by the network and more context tokens are generated, leading to more computational cost in the proposed CFM module. For fairness, when studying the impact of r, we keep other parameters unchanged. The ablation study is conducted on the VSPW [17] validation set.

The results are shown in Tab. 6. We have several observations. First, when using small receptive fields, our method achieves the mIoU of 37.2%, which is already better than the baseline (SegFormer [52]) with the mIoU of 36.5%. Second, when increasing the receptive fields so that the model can see larger regions in farther frames, the performance significantly improves, from 37.2% to 39.2%, implying the value of static and motional contexts. Third, further increasing the receptive fields to $\{49, 35, 21, 7\}$ does not boost performance. The possible reason is that when receptive fields become large enough, no further useful contexts can be exploited. In general, using a reasonable r gives good performance and our method is robust to the reasonable choice of the receptive fields r.

Impact on the Pooling Windows. To investigate the impact of pooling kernels/windows, we conduct ablation study on different $p = \{p_{t-k_1}, p_{t-k_2}, p_{t-k_3}, p_t\}$, where $k_1 = 9$, $k_2 = 6$, and $k_3 = 3$, following [17]. While ablating p, we keep other parameters the same for fair comparisons. Note that a smaller pooling window indicates more fine-grained features are extracted, and hence more context tokens are generated, leading to more computational cost in our multi-head non-self attention layer. The ablation study is conducted on the VSPW [17] validation set.

The results are shown in Tab. 7. First, for different choices of pooling windows p, our method is much better than the SegFormer [52] baseline with mIoU of 36.5%. When increasing the granularity (more fine-grained features are exploited) from $\{7, 7, 7, 1\}$ to $\{7, 5, 3, 1\}$, and to $\{5, 3, 3, 1\}$, better mIoU scores are obtained, *i.e.*, from 38.3% to 38.5%, and to 38.7%. In general, our method is robust to the choice of p and the reasonable p gives a good performance.

TABLE 6
Ablation study on the impact of the receptive fields. The used backbone is MiT-B1. The proposed method is robust to reasonable receptive fields.

r	mIoU	mVC_8	mVC_{16}
{7,5,3,7}	37.2	88.0	83.4
{21,15,9,7}	38.0	88.2	83.6
{35,15,9,7}	38.3	88.2	83.7
{49,15,9,7}	38.7	88.0	83.3
{49,20,6,7}	38.5	88.6	84.1
{49,25,15,7}	**39.2**	**88.6**	**84.1**
{49,35,21,7}	38.8	88.6	84.1

TABLE 7
Ablation study on the impact of the pooling windows. The used backbone is MiT-B1. The proposed method is robust to the choice of pooling windows.

p	mIoU	mVC_8	mVC_{16}
{5,3,3,1}	**38.7**	88.2	83.7
{7,5,3,1}	38.5	**88.6**	**84.1**
{7,7,7,1}	38.3	88.3	83.8

Ablation on local and global temporal contexts. In this experiment, we study the impact of local temporal contexts (static and motional contexts) and global temporal contexts on performance. Different from previous methods, the proposed CFFM can learn both static and motional contexts (local temporal contexts) in a unified model. When CFFM predicts the segmentation mask for the current frame, it uses three previous frames as reference frames. Based on CFFM, we further propose CFFM++ which extends CFFM by further adding global temporal contexts. The results for this ablation study are shown in Tab. 8. We start from "Baseline" method (A) which means SegFormer with MiT-B1 backbone. We simulate a case where only static contexts can be used, by replicating the current frame three times and using them as the reference frames. In this way, only static contexts could be used since all the reference frames are the same as the current one. We denote this experiment as "Baseline+static contexts" (B). We also conduct experiments on only adding global temporal contexts on "Baseline", which leads to D. By adding global temporal contexts on CFFM (C), we obtain our full model CFFM++ (F). Following our setting in ablation studies, we use the VSPW val dataset.

From the table, it can be seen that by adding static contexts to the baseline (A), a mIoU gain of 2.2 is achieved. However, this variant (B) does not show improvements in temporal consistency metrics mVC8 and mVC16 since no temporal information from neighboring frames is used. When adding static/motional contexts to the baseline, the method is CFFM (C) and achieves performance gains in all metrics mIoU, mVC_8 and mVC_{16}. By further adding global temporal contexts on top of CFFM, we get our full model CFFM++ (F) which outperforms CFFM, showing the effectiveness of global temporal contexts. By comparing D and A, we can also see the power of global temporal contexts.

Ablation on the number of frames. For CFFM, the number of frames being used to extract local temporal contexts is $N_f = l+1$, where l is the number of reference frames as introduced in §3. To fairly study the impact of the number of frames, we also set the hyperparameter f (§4) for extracting global temporal contexts to

TABLE 8
Ablation study on local temporal contexts (static and motional contexts) and global temporal contexts.

Symbol	Methods	mIoU	mVC_8	mVC_{16}
A	Baseline	36.5	84.7	79.9
B	Baseline+static contexts	37.7	84.4	79.4
C	Baseline+static/motional contexts (CFFM)	38.5	88.6	84.1
D	Baseline+global temporal contexts	38.4	85.0	80.3
E	Baseline+static contexts+global temporal contexts	39.2	85.5	80.8
F	Baseline+static/motional contexts +global temporal contexts (CFFM++)	**39.9**	**89.1**	**84.9**

TABLE 9
Ablation study on the number of input frames.

N_f	Methods	mIoU	mVC_8	mVC_{16}
4	CFFM	38.5	88.6	84.1
	CFFM+	38.0	84.9	80.1
	CFFM++	**39.6**	**88.8**	**84.6**
6	CFFM	38.8	89.1	84.4
	CFFM+	38.2	85.1	80.2
	CFFM++	**40.0**	**89.4**	**85.3**

TABLE 10
Ablation study on the number (N_p) of extracted prototypes. We use MiT-B1 as the backbone.

Methods	N_p	mIoU	mVC_8	mVC_{16}
CFFM (Ours)	-	38.5	88.6	84.1
CFFM++ (Ours)	10	39.5	88.7	84.5
	50	39.8	89.0	84.8
	100	**39.9**	**89.1**	**84.9**
	200	39.7	89.0	84.8

TABLE 11
Study on effectiveness of prototypes. We use prototypes extracted from different backbone models for CFFM++ (MiT-B0).

Methods	Prototypes Model	mIoU	mVC_8	mVC_{16}
CFFM++ (MiT-B0)	MiT-B0	35.9	88.4	83.8
	MiT-B1	38.0	88.7	84.6
	MiT-B2	39.1	89.1	85.1
	MiT-B5	**39.6**	**89.3**	**85.4**

be l. Hence, the number of frames being used for local and global temporal contexts is the same. In this part, we ablate on N_f. We also show the result of a variant *CFFM+*, which *only* uses global temporal contexts. The results are shown in Tab. 9.

From the table, it can be seen that when using the same number of frames for extracting local and global temporal contexts, CFFM++ always outperforms CFFM. This is due to the fact that CFFM++ (using local and global temporal contexts) is built on top of CFFM (using local temporal contexts). Comparing CFFM+ and CFFM, we can observe CFFM is slightly better than CFFM+, since local temporal contexts are more informative than the global temporal contexts. The fact that CFFM++ outperforms both CFFM and CFFM+ shows that local temporal contexts and global temporal contexts are complementary.

Influence of the number of global temporal contextual prototypes. In our experiments, we set the number (N_p) of contextual prototypes as 100 when extracting global temporal information. Here, we study the influence of this parameter. The results are shown in Tab. 10. we observe that compressing the global temporal contexts into only 10 prototypes already gives promising results. It demonstrates the effectiveness of the global temporal contexts and that the information from faraway video frames can provide additional guidance to help segment the target frame. When increasing the number of generated prototypes from 10 to 100, better performance can be achieved due to the fact that more detailed contexts are extracted and exploited. However, further extracting more (*e.g.*, 200) contextual prototypes does not help. For a certain video, a good number (*e.g.*, 100) of prototypes could already represent the existing contextual information well. Further increasing the number of prototypes will not significantly include more information. This is consistent with the discovery in the few-shot semantic segmentation paper ASGNet [110].

Knowledge distillation of global temporal contextual prototypes. Here, we conduct knowledge distillation experiments using global temporal contextual prototypes. Specifically, the prototypes (with size of $\mathbb{R}^{100\times256}$) extracted from large models (MiT-B5, MiT-B2, MiT-b1) are used by CFFM++ on small backbone (MiT-B0). This study can be interpreted from the perspective of knowledge distillation. We distill knowledge from large models to the small model, through the extracted contextual prototypes. The results are shown in Tab. 11. It can be observed that by simply replacing the prototypes from MiT-B0 with the prototypes from MiT-B1, MiT-B2, and MiT-B5, significant performance improvements are obtained for CFFM++ (MiT-B0), which demonstrates the extracted prototypes contain rich contextual information.

6 Conclusion and Future Work

The video contexts contain *local temporal contexts* which represent the contextual information from neighbouring/nearby frames and *global temporal contexts* which indicate the contexts from the whole video. This paper first studies local temporal contexts which can be further divided into *static contexts* and *motional contexts* within the nearby frames. Previous methods pay much attention to motional contexts but ignore the static contexts. We propose a Coarse-to-Fine Feature Mining (CFFM) technique to jointly learn a unified presentation of static and motional contexts, for precise and efficient VSS. CFFM contains two parts: Coarse-to-Fine Feature Assembling (CFFA) and Cross-frame Feature Mining (CFM). The former summarizes contextual information with different granularity for different frames, according to their distance to the target frame. The latter efficiently mines the contexts from neighbouring frames to enhance the feature of the target frame. To make use of *global temporal contexts*, we further propose CFFM++ which abstracts global temporal contextual prototypes from the video by unsupervised clustering and then exploits them to improve the target frame features. Extensive experiments show that CFFM boosts segmentation performance in a clear margin while adding limited computational cost. What's more, CFFM++ clearly surpasses CFFM with the help of global temporal information.

For future work, in addition to the above two aspects (local and global temporal contexts), the following two directions are promising. First, our exploration of contextual information for VSS focuses on simultaneously learning temporal contexts for all semantic categories. Considering the relationships amongst various categories (*e.g.*, *horses* is often related to the *grassland*), the explicit modeling of class-specific temporal contexts is also an interesting direction to explore. Second, it would be also interesting to extend our methods to other video tasks that require the learning of temporal contexts.

REFERENCES

[1] G. Sun, Y. Liu, H. Ding, T. Probst, and L. Van Gool, "Coarse-to-fine feature mining for video semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 3126–3137. 1, 2

[2] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, 2017. 1, 3, 4, 8, 9

[3] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, 2017. 1, 3

[4] Z. Tian, T. He, C. Shen, and Y. Yan, "Decoders matter for semantic segmentation: Data-dependent decoding enables flexible feature aggregation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 3126–3135. 1, 3

[5] J. Liu, J. He, J. Zhang, J. S. Ren, and H. Li, "EfficientFCN: Holistically-guided decoding for semantic segmentation," in *Eur. Conf. Comput. Vis.*, 2020, pp. 1–17. 1, 3

[6] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, 2015. 1

[7] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes dataset for semantic urban scene understanding," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 3213–3223. 1, 3, 7, 8, 9

[8] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ADE20K dataset," *Int. J. Comput. Vis.*, vol. 127, no. 3, pp. 302–321, 2019. 1

[9] H. Caesar, J. Uijlings, and V. Ferrari, "COCO-Stuff: Thing and stuff classes in context," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 1209–1218. 1

[10] S. Waqas Zamir, A. Arora, A. Gupta, S. Khan, G. Sun, F. Shahbaz Khan, F. Zhu, L. Shao, G.-S. Xia, and X. Bai, "iSAID: A large-scale dataset for instance segmentation in aerial images," in *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2019, pp. 28–37. 1

[11] R. Gadde, V. Jampani, and P. V. Gehler, "Semantic video CNNs through representation warping," in *Int. Conf. Comput. Vis.*, 2017, pp. 4453–4462. 1, 3, 4

[12] X. Jin, X. Li, H. Xiao, X. Shen, Z. Lin, J. Yang, Y. Chen, J. Dong, L. Liu, Z. Jie *et al.*, "Video scene parsing with predictive feature learning," in *Int. Conf. Comput. Vis.*, 2017, pp. 5580–5588. 1, 3

[13] S. Liu, C. Wang, R. Qian, H. Yu, R. Bao, and Y. Sun, "Surveillance video parsing with single frame supervision," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 413–421. 1, 3, 4

[14] D. Nilsson and C. Sminchisescu, "Semantic video segmentation by gated recurrent flow propagation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 6819–6828. 1, 3, 4, 9

[15] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Eur. Conf. Comput. Vis.*, 2012. 1, 3

[16] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *Eur. Conf. Comput. Vis.*, 2008, pp. 44–57. 1, 3

[17] J. Miao, Y. Wei, Y. Wu, C. Liang, G. Li, and Y. Yang, "VSPW: A large-scale dataset for video scene parsing in the wild," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 4133–4143. 1, 2, 3, 4, 6, 7, 8, 9, 10, 11

[18] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 7151–7160. 1, 2, 3, 4

[19] Z. Jin, T. Gong, D. Yu, Q. Chu, J. Wang, C. Wang, and J. Shao, "Mining contextual information beyond image for semantic segmentation," in *Int. Conf. Comput. Vis.*, 2021, pp. 7231–7241. 1, 3, 4

[20] Z. Jin, B. Liu, Q. Chu, and N. Yu, "ISNet: Integrate image-level and semantic-level context for semantic segmentation," in *Int. Conf. Comput. Vis.*, 2021, pp. 7189–7198. 1, 3, 4

[21] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *Eur. Conf. Comput. Vis.*, 2020, pp. 173–190. 1, 3, 4, 8

[22] J. Liu, J. He, Y. Qiao, J. S. Ren, and H. Li, "Learning to predict context-adaptive convolution for semantic segmentation," in *Eur. Conf. Comput. Vis.*, 2020, pp. 769–786. 1, 3, 4

[23] Y. Zhou, X. Sun, Z.-J. Zha, and W. Zeng, "Context-reinforced semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 4046–4055. 1, 3, 4

[24] J. He, Z. Deng, L. Zhou, Y. Wang, and Y. Qiao, "Adaptive pyramid context network for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 7519–7528. 1, 2, 3, 4

[25] H. Ding, X. Jiang, B. Shuai, A. Q. Liu, and G. Wang, "Context contrasted feature and gated multi-scale aggregation for scene segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 2393–2402. 1

[26] X. Li, Y. Yang, Q. Zhao, T. Shen, Z. Lin, and H. Liu, "Spatial pyramid based graph reasoning for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 8950–8959. 1, 3, 4

[27] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2018. 1, 2, 3, 4

[28] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Eur. Conf. Comput. Vis.*, 2018, pp. 801–818. 1, 2, 3, 4, 8

[29] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 2881–2890. 1, 2, 3, 4, 8, 9

[30] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "DenseASPP for semantic segmentation in street scenes," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 3684–3692. 1, 2, 3, 4

[31] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "CCNet: Criss-cross attention for semantic segmentation," in *Int. Conf. Comput. Vis.*, 2019, pp. 603–612. 1, 2, 3, 4

[32] Z. Zhu, M. Xu, S. Bai, T. Huang, and X. Bai, "Asymmetric non-local neural networks for semantic segmentation," in *Int. Conf. Comput. Vis.*, 2019, pp. 593–602. 1, 2, 3, 4

[33] M. Zhen, J. Wang, L. Zhou, S. Li, T. Shen, J. Shang, T. Fang, and L. Quan, "Joint semantic segmentation and boundary detection using iterative pyramid contexts," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 13 666–13 675. 1, 3, 4

[34] Y.-S. Xu, T.-J. Fu, H.-K. Yang, and C.-Y. Lee, "Dynamic video segmentation network," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 6556–6565. 1, 3, 4, 9

[35] Y. Liu, C. Shen, C. Yu, and J. Wang, "Efficient semantic video segmentation with per-frame inference," in *Eur. Conf. Comput. Vis.*, 2020, pp. 352–368. 1, 3, 4, 8, 9

[36] P. Hu, F. Caba, O. Wang, Z. Lin, S. Sclaroff, and F. Perazzi, "Temporally distributed networks for fast video semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 8818–8827. 1, 3

[37] E. Shelhamer, K. Rakelly, J. Hoffman, and T. Darrell, "Clockwork convnets for video semantic segmentation," in *Eur. Conf. Comput. Vis.*, 2016, pp. 852–868. 1, 3, 4, 9

[38] S. Jain, X. Wang, and J. E. Gonzalez, "Accel: A corrective fusion network for efficient semantic segmentation on video," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 8866–8875. 1, 3, 4, 9

[39] J. Carreira, V. Patraucean, L. Mazare, A. Zisserman, and S. Osindero, "Massively parallel video networks," in *Eur. Conf. Comput. Vis.*, 2018, pp. 649–666. 1, 3, 4

[40] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei, "Deep feature flow for video recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 2349–2358. 1, 3, 4, 9

[41] Y. Li, J. Shi, and D. Lin, "Low-latency video semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 5997–6005. 1, 3, 4

[42] S.-P. Lee, S.-C. Chen, and W.-H. Peng, "GSVNet: Guided spatially-varying convolution for fast semantic segmentation on video," in *IEEE Int. Conf. Multimedia Expo*, 2021, pp. 1–6. 1, 3, 4

[43] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Int. Conf. Comput. Vis.*, 2015, pp. 2758–2766. 2

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Annu. Conf. Neur. Inform. Process. Syst.*, 2017, pp. 6000–6010. 2, 4, 6

[45] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Int. Conf. Learn. Represent.*, 2021. 2, 3, 4, 6

[46] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 7794–7803. 2, 4, 6

[47] W. Xu, Y. Xu, T. Chang, and Z. Tu, "Co-Scale conv-attentional image transformers," in *Int. Conf. Comput. Vis.*, 2021, pp. 9981–9990. 2, 4

[48] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Int. Conf. Comput. Vis.*, 2021, pp. 10 012–10 022. 2, 3, 4, 5, 6

[49] B. Heo, S. Yun, D. Han, S. Chun, J. Choe, and S. J. Oh, "Rethinking spatial dimensions of vision transformers," in *Int. Conf. Comput. Vis.*, 2021, pp. 11 936–11 945. 2, 4

[50] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer, "Multiscale vision transformers," in *Int. Conf. Comput. Vis.*, 2021, pp. 6824–6835. 2, 4

[51] X. Chu, Z. Tian, Y. Wang, B. Zhang, H. Ren, X. Wei, H. Xia, and C. Shen, "Twins: Revisiting the design of spatial attention in vision transformers," in *Annu. Conf. Neur. Inform. Process. Syst.*, 2021. 2, 4

[52] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Annu. Conf. Neur. Inform. Process. Syst.*, 2021. 2, 3, 7, 8, 9, 10, 11

[53] Y. Zhang, S. Borse, H. Cai, and F. Porikli, "Auxadapt: Stable and efficient test-time adaptation for temporally consistent video semantic segmentation," in *IEEE Winter Conf. App. Comput. Vis.*, 2022, pp. 2339–2348. 2

[54] D. Zhang, H. Zhang, J. Tang, X.-S. Hua, and Q. Sun, "Self-regulation for semantic segmentation," in *Int. Conf. Comput. Vis.*, 2021, pp. 6953–6963. 3

[55] C.-W. Hsiao, C. Sun, H.-T. Chen, and M. Sun, "Specialize and fuse: Pyramidal output representation for semantic segmentation," in *Int. Conf. Comput. Vis.*, 2021, pp. 7137–7146. 3

[56] L. Zhu, D. Ji, S. Zhu, W. Gan, W. Wu, and J. Yan, "Learning statistical texture for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 12 537–12 546. 3

[57] M. Liu, D. Schonfeld, and W. Tang, "Exploit visual dependency relations for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 9726–9735. 3

[58] L. Wang, D. Li, Y. Zhu, L. Tian, and Y. Shan, "Dual super-resolution learning for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 3774–3783. 3

[59] Y. Pang, Y. Li, J. Shen, and L. Shao, "Towards bridging semantic gap to improve semantic segmentation," in *Int. Conf. Comput. Vis.*, 2019, pp. 4230–4239. 3

[60] Y. Nirkin, L. Wolf, and T. Hassner, "HyperSeg: Patch-wise hypernetwork for real-time semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 4061–4070. 3

[61] H. Hu, D. Ji, W. Gan, S. Bai, W. Wu, and J. Yan, "Class-wise dynamic graph convolution for semantic segmentation," in *Eur. Conf. Comput. Vis.*, 2020, pp. 1–17. 3

[62] W. Chen, X. Zhu, R. Sun, J. He, R. Li, X. Shen, and B. Yu, "Tensor low-rank reconstruction for semantic segmentation," in *Eur. Conf. Comput. Vis.*, 2020, pp. 52–69. 3

[63] H. Zhang, H. Zhang, C. Wang, and J. Xie, "Co-occurrent features in semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 548–557. 3

[64] Z. Wei, J. Zhang, L. Liu, F. Zhu, F. Shen, Y. Zhou, S. Liu, Y. Sun, and L. Shao, "Building detail-sensitive semantic segmentation networks with polynomial pooling," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 7115–7123. 3

[65] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Learning a discriminative feature network for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 1857–1866. 3

[66] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-SCNN: Gated shape CNNs for semantic segmentation," in *Int. Conf. Comput. Vis.*, 2019, pp. 5229–5238. 3

[67] Y. Liu, M.-M. Cheng, D.-P. Fan, L. Zhang, J.-W. Bian, and D. Tao, "Semantic edge detection with diverse deep supervision," *Int. J. Comput. Vis.*, vol. 130, no. 1, pp. 179–198, 2022. 3

[68] X. Li, X. Li, L. Zhang, G. Cheng, J. Shi, Z. Lin, S. Tan, and Y. Tong, "Improving semantic segmentation via decoupled body and edge supervision," in *Eur. Conf. Comput. Vis.*, 2020, pp. 435–452. 3

[69] C. Wang, Y. Zhang, M. Cui, J. Liu, P. Ren, Y. Yang, X. Xie, X. Hua, H. Bao, and W. Xu, "Active boundary loss for semantic segmentation," in *AAAI Conf. Artif. Intell.*, 2022, pp. 2397–2405. 3

[70] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 3640–3649. 3

[71] X. Li, Z. Zhong, J. Wu, Y. Yang, Z. Lin, and H. Liu, "Expectation-maximization attention networks for semantic segmentation," in *Int. Conf. Comput. Vis.*, 2019, pp. 9167–9176. 3

[72] F. Zhang, Y. Chen, Z. Li, Z. Hong, J. Liu, F. Ma, J. Han, and E. Ding, "ACFNet: Attentional class feature network for semantic segmentation," in *Int. Conf. Comput. Vis.*, 2019, pp. 6798–6807. 3

[73] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 3146–3154. 3

[74] S. Seifi and T. Tuytelaars, "Attend and segment: Attention guided active semantic segmentation," in *Eur. Conf. Comput. Vis.*, 2020, pp. 305–321. 3

[75] Z. Zhong, Z. Q. Lin, R. Bidart, X. Hu, I. B. Daya, Z. Li, W.-S. Zheng, J. Li, and A. Wong, "Squeeze-and-attention networks for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 13 065–13 074. 3

[76] H. Ding, X. Jiang, B. Shuai, A. Q. Liu, and G. Wang, "Semantic correlation promoted shape-variant context for segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 8885–8894. 3

[77] J. He, Z. Deng, and Y. Qiao, "Dynamic multi-scale filters for semantic segmentation," in *Int. Conf. Comput. Vis.*, 2019, pp. 3562–3572. 3, 4

[78] A. Kundu, V. Vineet, and V. Koltun, "Feature space optimization for semantic video segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 3168–3175. 3, 4

[79] B. Mahasseni, S. Todorovic, and A. Fern, "Budget-aware deep semantic video segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 1029–1038. 3, 4

[80] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Annu. Conf. Neur. Inform. Process. Syst.*, 2014. 3

[81] Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. Newsam, A. Tao, and B. Catanzaro, "Improving semantic segmentation via video propagation and label relaxation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 8856–8865. 3

[82] J. Li, W. Wang, J. Chen, L. Niu, J. Si, C. Qian, and L. Zhang, "Video semantic segmentation via sparse temporal transformer," in *ACM Int. Conf. Multimedia*, 2021, pp. 59–68. 3

[83] M. Paul, M. Danelljan, L. Van Gool, and R. Timofte, "Local memory attention for fast video semantic segmentation," in *Int. Conf. Intell. Robot. Syst.*, 2021, pp. 1102–1109. 3

[84] H. Ding, C. Liu, S. He, X. Jiang, and C. C. Loy, "MeViS: A large-scale benchmark for video segmentation with motion expressions," in *Int. Conf. Comput. Vis.*, 2023. 3

[85] J. Yang, C. Li, P. Zhang, X. Dai, B. Xiao, L. Yuan, and J. Gao, "Focal self-attention for local-global interactions in vision transformers," *arXiv preprint arXiv:2107.00641*, 2021. 3, 4, 5

[86] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "SwinIR: Image restoration using swin transformer," in *Int. Conf. Comput. Vis.*, 2021, pp. 1833–1844. 3

[87] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z.-H. Jiang, F. E. Tay, J. Feng, and S. Yan, "Tokens-to-token ViT: Training vision transformers from scratch on ImageNet," in *Int. Conf. Comput. Vis.*, 2021, pp. 558–567. 3

[88] Y. Liu, Y.-H. Wu, G. Sun, L. Zhang, A. Chhatkuli, and L. Van Gool, "Vision transformers with hierarchical attention," *arXiv preprint arXiv:2106.03180*, 2021. 3

[89] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr, and L. Zhang, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 6881–6890. 3, 7

[90] H. Ding, C. Liu, S. He, X. Jiang, P. H. Torr, and S. Bai, "MOSE: A new dataset for video object segmentation in complex scenes," in *Int. Conf. Comput. Vis.*, 2023. 3

[91] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou, "PoinTr: Diverse point cloud completion with geometry-aware transformers," in *Int. Conf. Comput. Vis.*, 2021, pp. 12 498–12 507. 3

[92] G. Sun, Y. Liu, T. Probst, D. P. Paudel, N. Popovic, and L. V. Gool, "Rethinking global context in crowd counting," *Machine Intelligence Research*, 2023. 3

[93] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, "Transformer tracking," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 8126–8135. 3

[94] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu, "Learning spatio-temporal transformer for visual tracking," in *Int. Conf. Comput. Vis.*, 2021, pp. 10 448–10 457. 3

[95] D. Liang, X. Chen, W. Xu, Y. Zhou, and X. Bai, "TransCrowd: Weakly-supervised crowd counting with transformers," *Sci. China Inform. Sci.*, vol. 65, no. 6, pp. 1–14, 2022. 3

[96] J. Lanchantin, T. Wang, V. Ordonez, and Y. Qi, "General multi-label image classification with transformers," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 16 478–16 488. 3

[97] H. Wang, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen, "Max-DeepLab: End-to-end panoptic segmentation with mask transformers," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 5463–5474. 3

[98] H. Ding, C. Liu, S. Wang, and X. Jiang, "VLT: Vision-language transformer and query generation for referring segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 7900–7916, 2023. 3

[99] Z. Li, W. Wang, E. Xie, Z. Yu, A. Anandkumar, J. M. Alvarez, P. Luo, and T. Lu, "Panoptic segformer: Delving deeper into panoptic segmentation with transformers," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 1280–1289. 3

[100] B. Cheng, A. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," in *Annu. Conf. Neur. Inform. Process. Syst.*, 2021, pp. 17 864–17 875. 3

[101] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 1290–1299. 3

[102] Z. Chen, Y. Duan, W. Wang, J. He, T. Lu, J. Dai, and Y. Qiao, "Vision transformer adapter for dense predictions," in *Int. Conf. Learn. Represent.*, 2023. 3

[103] T. Zhang, X. Tian, H. Wei, Y. Wu, S. Ji, X. Wang, Y. Zhang, and P. Wan, "1st place solution for pvuw challenge 2023: Video panoptic segmentation," *arXiv preprint arXiv:2306.04091*, 2023. 3

[104] J. Su, W. Yang, J. Luo, and X. Wei, "3rd place solution for pvuw challenge 2023: Video panoptic segmentation," *arXiv preprint arXiv:2306.06753*, 2023. 3

[105] G. Sun, Y. Liu, H. Tang, A. Chhatkuli, L. Zhang, and L. Van Gool, "Mining relations among cross-frame affinities for video semantic segmentation," in *European Conference on Computer Vision*. Springer, 2022, pp. 522–539. 4

[106] H. Ding, X. Jiang, A. Q. Liu, N. M. Thalmann, and G. Wang, "Boundary-aware feature propagation for scene segmentation," in *Int. Conf. Comput. Vis.*, 2019, pp. 6819–6829. 4

[107] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Int. Conf. Comput. Vis.*, 2021, pp. 568–578. 4, 5

[108] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Annu. Conf. Neur. Inform. Process. Syst.*, 2019, pp. 8026–8037. 5

[109] L. Yang, W. Zhuo, L. Qi, Y. Shi, and Y. Gao, "Mining latent classes for few-shot segmentation," in *Int. Conf. Comput. Vis.*, 2021, pp. 8721–8730. 6

[110] G. Li, V. Jampani, L. Sevilla-Lara, D. Sun, J. Kim, and J. Kim, "Adaptive prototype learning and allocation for few-shot segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 8334–8343. 6, 12

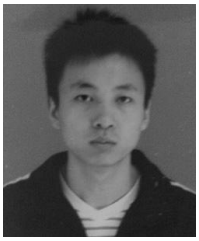
[111] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *Eur. Conf. Comput. Vis.*, 2018, pp. 418–434. 8

[112] M. Contributors, "MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark," https://github.com/open-mmlab/mmsegmentation, 2020. 7

[113] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015. 7

[114] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in *Annu. Conf. Neur. Inform. Process. Syst.*, 2011, pp. 109–117. 7

Guolei Sun received his Ph.D. degree at ETH Zurich, Switzerland, in Prof. Luc Van Gool's Computer Vision Lab in Jan 2024. Before that, he got master degree in computer science from the King Abdullah University of Science and Technology (KAUST), in 2018. He is currently a postdoctoral researcher at Computer Vision Lab, ETH Zurich. From 2018 to 2019, he worked as a research engineer with the Inception Institute of Artificial Intelligence, UAE. His research interests include deep learning for video understanding, semantic/instance segmentation, object counting, and weakly supervised learning.



Yun Liu received his B.E. and Ph.D. degrees from Nankai University in 2016 and 2020, respectively. Then, he worked with Prof. Luc Van Gool as a postdoctoral scholar at Computer Vision Lab, ETH Zurich, Switzerland. Currently, he is a senior scientist at the Institute for Infocomm Research (I2R), A*STAR, Singapore. His research interests include computer vision and machine learning.

Henghui Ding received his B.E. degree from Xi'an Jiaotong University, Xi'an, China, in 2016. He received the Ph.D. degree from Nanyang Technological University (NTU), Singapore, in 2020. He was a Postdoctoral Researcher at the Computer Vision Lab of ETH Zurich in Switzerland and a Research Scientist at ByteDance AI Lab in Singapore. He is currently a Presidential Postdoctoral Fellow (Principal Investigator) at NTU. He serves as Associate Editors for IET Computer Vision and Visual Intelligence. He serves/served as Area Chair for CVPR'24 and ACM MM'24, and Senior Program Committee member for AAAI'(22-24) and IJCAI'(23-24). His research interests include computer vision and machine learning.

Min Wu (Senior Member, IEEE) received the B.E. degree in computer science from USTC, China, in 2006, and the Ph.D. degree in computer science from NTU, Singapore, in 2011. He is currently a principal scientist with the Institute for Infocomm Research (I2R), A*STAR, Singapore. He received the best paper awards in the IEEE ICIEA 2022, the IEEE SmartCity 2022, the InCoB 2016, and the DASFAA 2015. He also won the CVPR UG2+ challenge in 2021 and the IJCAI competition on repeated buyers prediction in 2015. His current research interests include machine learning, data mining, and bioinformatics.

Luc Van Gool received the degree in electromechanical engineering from the Katholieke Universiteit Leuven, in 1981. Currently, he is a professor at the Katholieke Universiteit Leuven in Belgium and the ETH in Zurich, Switzerland. He leads computer vision research with both places and also teaches at both. He has been a program committee member of several major computer vision conferences. His main research interests include 3D reconstruction and modeling, object recognition, tracking, gesture analysis, and a combination of those. He received several Best Paper awards, won a David Marr Prize and a Koenderink Award, and was nominated Distinguished Researcher by the IEEE Computer Science committee. He is a co-founder of 12 spin-off companies.