Dual-Temporal Exemplar Representation Network for Video Semantic Segmentation

Xiaolong Xu[1], Lei Zhang[1,2]*, Jiayi Li[1], Lituan Wang[1], Yifan Guan[1], Yu Yan[1], Leyi Zhang[1], Hao Song[1]
[1]Sichuan University, [2]Tianfu Jincheng Laboratory
xilo@stu.scu.edu.cn, leizhang@scu.edu.cn

Abstract

Video semantic segmentation aims to assign a class label for each pixel in every video frame. Existing methods predominantly follow the reference-target interaction paradigm, focusing on extracting local temporal contexts while neglecting the integration of global temporal information. Moreover, complex dynamics and varying lighting conditions introduce inter-frame intra-class discrepancies in feature representations, leading to unstable predictions. In this paper, we propose a novel framework, the Dual-Temporal Exemplar Representation Network (DTERN), which utilizes the strong representational capability of cluster centers, i.e., exemplars, to effectively model both local and global temporal information. DTERN consists of two core modules: 1) the Local Temporal Exemplar Module (LTEM), which constructs local exemplars to capture local temporal contexts, ensuring stable and reliable predictions. 2) the Global Temporal Exemplar Module (GTEM), which introduces learnable global exemplars to dynamically model global temporal information, thereby improving the effective consistency of segmentation. Furthermore, we observe that the existing Video Consistency (VC) metric fails to evaluate segmentation accuracy and lacks sensitivity to small-object segmentation. To this end, we propose Video Effective Consistency (VEC) to comprehensively evaluate temporal consistency and segmentation effectiveness. Experiments on VSPW and Cityscape demonstrate that DTERN outperforms state-of-the-art methods. The code is available at https://github.com/zlxilo/DTERN.

1. Introduction

Video semantic segmentation (VSS) involves assigning semantic labels to all pixels in a video. The core of VSS lies in effectively leveraging temporal information. Existing methods mainly focus on modeling the temporal relationship between reference and target frames through the reference-target interaction paradigm (Fig. 1a).

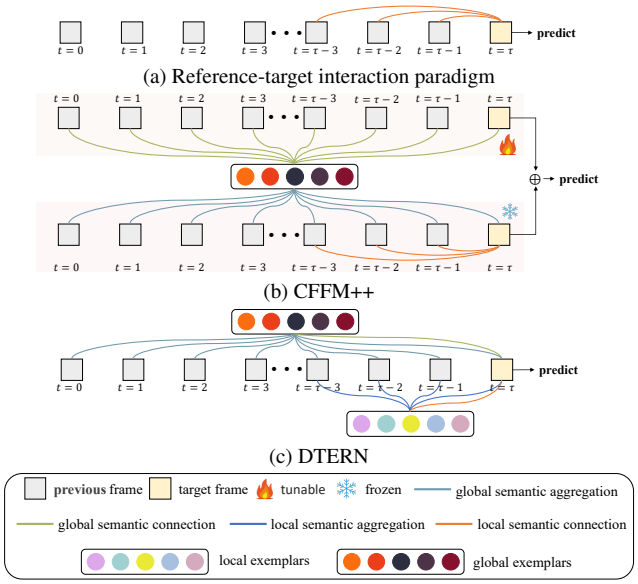


Figure 1. Comparison of existing VSS methods with DTERN. (a) Approaches using the reference-target interaction paradigm model only the local temporal information. (b) CFFM++ further models global temporal information by constructing non-learnable global exemplars. (c) DTERN models both local and global temporal information by constructs learnable exemplars.

Most of the existing VSS methods rely on optical flow to capture temporal relationships. Typically, optical flow is extracted by dedicated networks [7, 16] to warp the reference frame features into alignment with the target frame features, followed by a simple aggregation process [9, 14, 25, 35, 47]. Additionally, recent research [18, 31, 34, 41] has introduced the self-attention mechanism to model interactions between frames, enabling the target frame to be dynamically updated based on the reference frames. Meanwhile, researchers have explored approaches to achieve efficient and stable video semantic segmentation by reusing frame

*Corresponding author: Lei Zhang (leizhang@scu.edu.cn)

features [1, 21, 29] and improving reference frame selection strategies [38]. However, the above methods mainly focus on extracting local temporal contexts, while the utilization of global temporal information remains insufficient. To mine global temporal contexts, CFFM++ [33] constructs non-learnable global exemplars through a pre-trained encoder to model global temporal information, further improving segmentation performance (Fig. 1b). Nevertheless, CFFM++ is a multi-stage training framework that increases the training burden, while the effectiveness of the global temporal context mining with non-learnable exemplars is constrained by the performance of the pre-trained encoder. Moreover, these works fail to address the inter-frame intra-class discrepancies caused by complex factors such as lighting and occlusion, resulting in unstable predictions.

To address the above issues, we propose DTERN, which adopts a new interaction paradigm (Fig. 1c) that leverages the strong representational capability of cluster centers to model both local and global temporal information. Specifically, DTERN consists of two core modules: the Local Temporal Exemplar Module (LTEM) and the Global Temporal Exemplar Module (GTEM). The sequential LTEMs progressively enhance the target frame features, with each LTEM constructing local exemplars to model local temporal information. To achieve this, a Selection and Aggregation Module is designed to selectively construct and adaptively aggregate cluster centers, generating local exemplars within each LTEM, thereby mitigating prediction instability. Furthermore, to dynamically capture high-quality spatial and semantic information from each input frame, a Cluster Assignment Module is proposed to simulate the cluster assignment process. In GTEM, learnable tokens representing frame-level cluster centers (FCC) and video-level cluster centers (VCC) are introduced. During training, VCC progressively aggregates local temporal context from the updated target frame features to derive global temporal information, which is then used to further update the target frame features, improving the effective consistency of segmentation. Furthermore, we introduce the Video Effective Consistency (VEC) to address the limitations of the Video Consistency (VC) metric, which fails to evaluate segmentation accuracy and exhibits insufficient sensitivity to small-object segmentation.

In summary, our work has the following main contributions:

- To exploit temporal information and mitigate unstable predictions caused by inter-frame intra-class discrepancies, we propose DTERN to learn local and global temporal information by leveraging the strong representational capability of cluster centers.
- We propose Video Effective Consistency (VEC), a novel evaluation metric for video semantic segmentation. This dual-focus metric simultaneously assesses temporal consistency and segmentation effectiveness.
- We evaluate DTERN on the large-scale video semantic segmentation datasets VSPW and Cityscapes. DTERN outperforms several state-of-the-art methods.

2. Related Work

2.1. Image Semantic Segmentation

Image semantic segmentation is a core computer vision task that assigns semantic labels to every pixel in an image. While traditional methods relied on handcrafted features and shallow models, significant advancements in this field have been driven by Convolutional Neural Networks. Modern approaches [2, 3, 8, 12, 13, 15, 23, 26, 28, 42, 46, 48] leverage end-to-end deep neural networks for efficient scene segmentation and have achieved remarkable success. However, video data is more common in real-world scenarios, which has led to increasing research attention on VSS.

2.2. Video Semantic Segmentation

VSS performs pixel-level classification for each frame. Current approaches mainly model temporal relationships between reference and target frames, utilizing local temporal information to improve segmentation performance.

VSS approaches mostly utilize optical flow to capture temporal information. Specifically, NetWarp [9] and STTGU [25] employ feature warping to align features from reference frames to the target frame via optical flow and fuse them with the target frame features. ETC [22] uses optical flow to align the segmentation results of the reference and target frames and employs a temporal loss function to enforce consistency constraints. DFF [47] and Accel [17] propagate features from reference frames to the target frame via optical flow, reducing computational costs. Researchers also utilize self-attention mechanisms to model temporal information. Specifically, CFFM [31] introduces a coarse-to-fine feature mining method that learn both static and dynamic contextual information simultaneously. TMANet [34] uses self-attention mechanisms to capture the relationships between the current frame and the memorized frames. STT [18] dynamically captures the temporal relationships through a sparse self-attention mechanism. LMANet [27] enhances the segmentation performance by leveraging semantic information from past frames. Different from the above approaches, [1, 29] reuse high-level features directly. DVSNet [38] utilizes a dynamic key-frame selection strategy. CFFM++[33] generates global exemplars via clustering to enhance target frame features, improving the video semantic segmentation performance. Efficient VSS [17, 19, 38, 47] run costly networks only on reference frames (fine-tuning non-keyframe features), while High-Performance VSS [18, 22, 24, 31, 33] exploit temporal continuity to boost accuracy [11].

The above VSS approaches mainly follow the reference-target interaction paradigm for local temporal information mining. However, this paradigm fails to utilize global temporal information across entire videos. Though CFFM++ has learned global temporal information, the non-learnable global exemplars generated by the pre-trained encoder limit its performance. Moreover, intra-class discrepancies among frames may lead to inconsistent semantic propagation, resulting in unstable predictions. This paper proposes a novel end-to-end VSS framework that leverages the strong representational capability of cluster centers to capture local and global temporal information.

2.3. Difference with CFFM++

We observe that recent work, CFFM++ [33], also employs global exemplars to model global temporal information for VSS. However, there are key differences between CFFM++ and our approach. CFFM++ adopts a multi-stage training framework and models global temporal information using non-learnable global exemplars generated by a pre-trained encoder, which limits the model's ability. In contrast, DTERN models both local and global temporal contexts through trainable exemplars within an end-to-end training framework, providing greater flexibility and adaptability.

3. Methods

3.1. overview

As shown in Fig. 2a, DTERN includes a shared encoder, N LTEMs, a GTEM, and a segmentation head. Given a video clip with T frames $\left\{\mathbf{I}_{t_k} \in R^{3\times H\times W}\right\}_{k=1}^{T}$. The target frame is I_{t_T} and previous frames $\left\{\mathbf{I}_{t_k} \in R^{3\times H\times W}\right\}_{k=1}^{T-1}$ are reference frames. The encoder extracts multi-level features $\left\{\mathbf{Z}_{t_k}^{v} \in R^{C_v\times H_v\times W_v}\right\}_{v=1}^{V}$ for each frame. The convolutional layers then fuse these features to produce encoded outputs $\left\{\mathbf{F}_{t_k} \in R^{D\times H_t\times W_t}\right\}_{k=1}^{T}$. For the l-th LTEM, $\left\{\mathbf{F}_{t_k}^{l}\right\}_{k=1}^{T-1}$ and enhanced target frame features $\{\widetilde{\mathbf{F}_{t_T}^{l-1}} \in R^{D\times H_t\times W_t}\}$ from the last LTEM are fed into the module, which generates local exemplars $\{\widetilde{\mathbf{C}^{l}} \in R^{M\times D}\}$ and enhanced target frame features $\widetilde{\mathbf{F}_{t_T}^{l}}$. The GTEM enhances target frame features with global exemplars to improve temporal consistency and gets output $\{\widetilde{\mathbf{O}} \in R^{D\times H_t\times W_t}\}$. Finally, $\widetilde{\mathbf{O}}$ is processed through the segmentation head to produce the final segmentation result.

3.2. Local Temporal Exemplar Module

LTEM models local temporal information by constructing local exemplars and using them to enhance the target frame features. Specifically, LTEM consists of three submodules: Cluster Assignment Module (CAM), Selection and Aggregation Module (SAM), and Feature Enhancement Module (FEM). For the multi-frame input features of the l-th LTEM, the CAM generates cluster assignments $\mathbf{A}_{t_j}^{l}$ for each frame. Based on these assignments, the SAM constructs cluster centers for each frame and completes the formation of local exemplars $\widetilde{\mathbf{C}^{l}}$ by aggregating the cluster centers of the reference and target frames. The FEM leverages these local exemplars to enhance the target frame features.

CAM. As shown in Fig. 2b, CAM consists of two branches: Parallel Deep Convolutional Branch (PDCB) and Axial Decomposition and Correction Branch (ADCB). Specifically, due to intricate variations in illumination intensity, object morphology, and other factors within and across frames in video data, directly employing simply stacked convolutional layers is insufficient to capture high-quality texture features. To address this, the PDCB adopts a multi-branch design with parallel depthwise convolutions of varying kernel sizes. The extracted multi-scale features are aggregated to construct intra-frame texture feature. This process is formulated as:

$$L_{t_j}^{l} = \mathcal{F}_{\text{agg}}(\sum_{i=1}^{n} \text{DWConv}_{k_i}\left(x_{t_j}^{l}\right) + x_{t_j}^{l}) \tag{1}$$

where $\mathcal{F}_{\text{agg}}$ represents a point-wise convolutional layer, DWConv_{k_i} represents a depthwise convolution with a kernel size of k_i, and $x_{t_j}^{l}$ represents the output of the shallow convolutional layers processed from $F_{t_j}^{l}$.

The ADCB extracts spatial features via horizontal and vertical pooling. Then, small-kernel 1D convolution is applied along the orthogonal axis (*e.g.* vertical for horizontal pooling and vice versa) to refine the extracted features with small-scale axial corrections. The output at height h can be formalized as:

$$G_{t_j}^{l}(h) = \psi_{k_x\times 1}(\frac{1}{W}\sum_{i=0}^{W} y_{t_j}^{l}\,(h,i)) \tag{2}$$

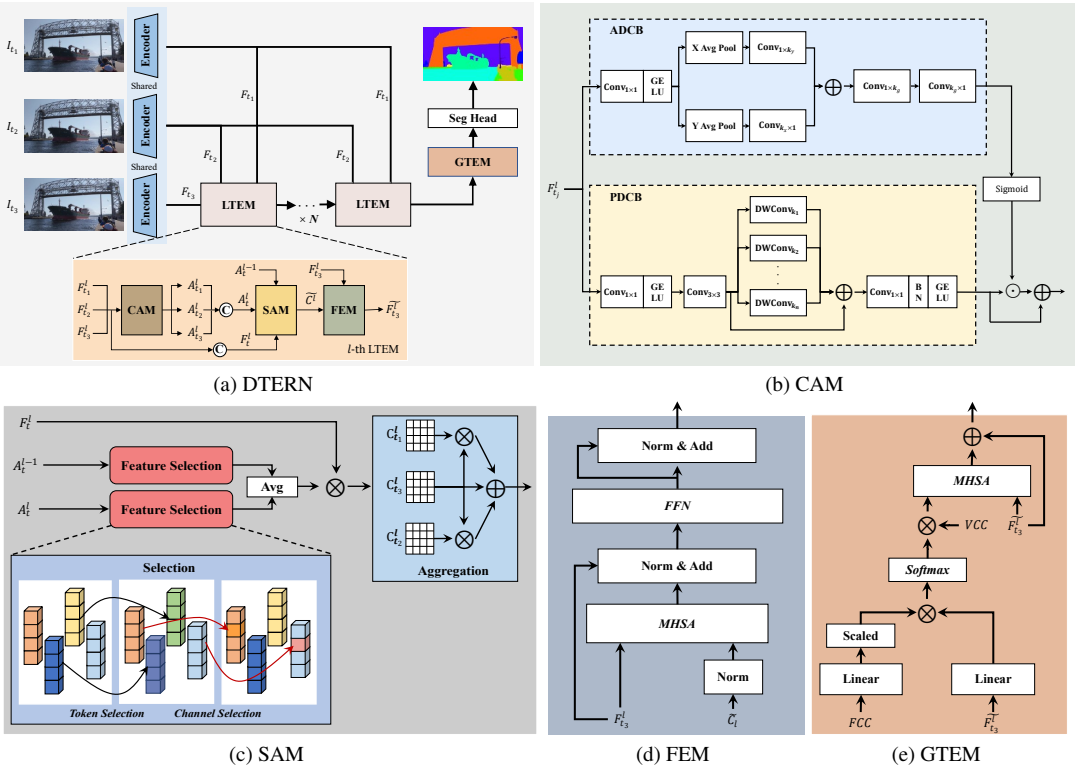
where $y_{t_j}^{l} = \psi_{1\times 1}(F_{t_j}^{l})$. Similarly, the output at width w can be formalized as:

$$G_{t_j}^{l}(w) = \psi_{1\times k_y}(\frac{1}{H}\sum_{i=0}^{H} y_{t_j}^{l}\,(w,i)) \tag{3}$$

After performing small-scale axial corrections separately along each axis, the features are aggregated by summation. Subsequently, two orthogonal 1D convolutions with larger kernels are applied to achieve large-scale edge corrections. This process can be expressed as:

$$G_{t_j}^{l} = \psi_{1\times k_g}(\psi_{k_g\times 1}(G_{t_j}^{l}(c,h) + G_{t_j}^{l}(c,w))) \tag{4}$$

By applying corrections to the feature map across multiple directions and scales, the model achieves enhanced accuracy of cluster assignment.



(a) DTERN (b) CAM

(c) SAM (d) FEM (e) GTEM

Figure 2. Overview of DTERN (three input frames illustrated, with I_{t_3} as the target frame). **(a) DTERN** consists of a shared encoder, N LTEMs, a GTEM, and a segmentation head. Each LTEM includes three submodules: CAM, SAM, and FEM. **(b) CAM** consists of PDCB and ADCB, which jointly perform cluster assignment for each frame. **(c) SAM** merges the cluster assignments through feature selection and aggregates the cluster centers of the reference and target frames to construct local exemplars. **(d) FEM** enhances the target frame features using local exemplars. **(e) GTEM** dynamically updates global exemplars to learn global temporal information.

A sigmoid function is applied to the ADCB output to generate a weight matrix, which is then used to enhance the texture features extracted by the PDCB. The final cluster assignment is obtained as:

$$\mathbf{A}^l_{t_j} = \psi_{1\times1}(L^l_{t_j} \odot \sigma(G^l_{t_j}) + L^l_{t_j}) \tag{5}$$

where σ denotes the sigmoid function, $\odot$ denotes the Hadamard product, $A^l_{t_j} \in R^{M\times H_t\times W_t}$ and M is the predefined number of clusters.

SAM. Considering the discrepancies in cluster assignments between adjacent LTEMs, we employ feature selection [30] to selectively merge these assignments. Specifically, we first concatenate all cluster assignments and flatten them along both temporal and spatial dimensions $A^l_t = flatten([A^l_{t_1}, A^l_{t_2}, \ldots, A^l_{t_T}])$. The selection operation is $\widetilde{A^l_t} = \phi \cdot A^l_t \cdot sim(A^l_t, \xi)$, where ξ represents the task embedding, and $sim(\cdot)$ denotes the cosine similarity function. We perform a mean operation on the cluster assignments of two adjacent layers to bring them closer together $\widehat{\mathbf{A}^l_t} = Avg(\widetilde{A^l_t}, \widetilde{A^{l-1}_t})$. Then, we reshape the result along the temporal dimension to recover the cluster assignments for each frame. Through matrix multiplication, we construct cluster centers for each frame as:

$$\mathbf{C}^l = \delta(\widehat{\mathbf{A}^l_t}) \otimes flatten(F^l_t)^\top \tag{6}$$

where F^l_t denotes the outcome of concatenating all frame features along the temporal dimension, the flattening operation is executed along the spatial dimension, δ is the softmax function and the cluster centers $\mathbf{C}^l \in R^{T\times M\times D}$.

As video data tends to be complex and variable, direct summing or averaging cluster centers across different frames may lead to erroneous feature alignment and unstable predictions. Video information can be divided into static regions, which capture consistent areas across consecutive frames, and dynamic regions, which highlight the content specific to the current frame. We hypothesize that the variations in cluster centers align with the inter-frame variations. Therefore, the local exemplars $\widetilde{\mathbf{C}^l}$ can be constructed by aggregating the cluster centers of the reference and target frames based on their similarity.

We define $w^l_{t_j}$ as the similarity between the cluster centers of the j-th frame and the target frame in the l-th LTEM,

while $\widetilde{\mathbf{C}^l}$ is denoted as:

$$\widetilde{\mathbf{C}^l} = \mathbf{C}^l_{t_T} + \sum_{j=0}^{T-1} w^l_{t_j} \mathbf{C}^l_{t_j} \tag{7}$$

where $\widetilde{\mathbf{C}^l} \in R^{M\times D}$ and $\mathbf{C}^l_{t_T}$ denotes the cluster centers of the target frame. We compute the similarity between cluster centers using the following formula:

$$w^l_{t_j} = \sigma(\partial + \beta cos(\mathbf{C}^l_{t_T}, \mathbf{C}^l_{t_j})) \tag{8}$$

where $cos(\cdot)$ denotes the cosine function, and ∂ and β are trainable parameters that are initialized to 0 and 1.

FEM. To enhance the features of the target frame with the local exemplars $\widetilde{\mathbf{C}^l}$, we first map the target frame features $F^l_{t_T}$ to a query $\{\boldsymbol{Q}^l_i \in R^{H_tW_t\times D}\}$ through the linear layer.

$$Q^l_i = FC(F^l_{t_T}, W_{query}) \tag{9}$$

where W_{query} represents the weights of the linear layer FC. Similarly, $\widetilde{\mathbf{C}^l}$ is mapped to key $\{\boldsymbol{K}^l_i \in R^{M\times D}\}$ and value $\{\boldsymbol{V}^l_i \in R^{M\times D}\}$ through the linear layer. Then, we update the target frame features using:

$$\widetilde{\mathbf{F}^l_{t_T}} = \delta\left(\frac{Q^l_i {K^l_i}^\top}{\sqrt{c}}\right) V_i + F^l_{t_T} \tag{10}$$

Since the predefined cluster number is typically smaller than the number of tokens ($M << H_tW_t$), our approach is expected to reduces the complexity from quadratic to linear.

Traditional methods for mining local temporal information predominantly rely on the reference-target interaction paradigm. However, they neglect inter-frame intra-class discrepancies caused by complex video variations, leading to unstable predictions. In contrast, we utilize local exemplars to model local temporal information, which can more effectively leverage consistent temporal information and enhance the accuracy and consistency of segmentation.

3.3. Global Temporal Exemplar Module

To mine global temporal information for improving effective consistency in segmentation, GTEM introduces learnable tokens: frame-wise cluster centers $\{FCC \in R^{T_g\times M_g\times D}\}$ and video-wise cluster centers $\{VCC \in R^{M_g\times D}\}$. FCC denotes the cluster centers of each frame input to the GTEM, interacting with the corresponding frame features to perform cluster assignments. VCC serves as the video-level cluster centers as well as the global exemplars. We leverage VCC to progressively model global temporal information from local temporal information contained in the updated target frame features. Specifically, we first apply linear transformations to the target frame features and FCC as follows:

$$d_x = FC(\widetilde{\mathbf{F}_{t_T}}, W_x) \quad d_y = FC(FCC, W_y) \tag{11}$$

To simulate cluster assignment, we calculate the affinity between FCC and the target frame features, subsequently deriving a probability distribution through softmax normalization as:

$$S_{x,y} = \delta(d_x \otimes d_y^\top) \tag{12}$$

We align the local temporal information with the global temporal information contained in VCC by matrix multiplication.

$$\widetilde{d_x} = S_{x,y} \otimes VCC \tag{13}$$

To strengthen the correlation between local and global temporal information, we further update the target frame features as:

$$\widetilde{\mathbf{O}} = \delta\left(\frac{\varphi_q(\widetilde{\mathbf{F}_{t_T}})\varphi_k^\top(\widetilde{d_x})}{\sqrt{c}}\right)\varphi_v(\widetilde{d_x}) + \widetilde{\mathbf{F}_{t_T}} \tag{14}$$

where $\varphi(\cdot)$ denotes a linear transformation operations. The $\widetilde{\mathbf{O}}$ will be fed into the segmentation head for processing to obtain the final segmentation result.

Through the utilization of global temporal information, GTEM improves the effective consistency of video semantic segmentation.

4. Experiment

4.1. Experimental Setup

Datasets. The proposed approach is primarily evaluated on VSPW [24], the largest dataset for video semantic segmentation. The dataset includes 2,806 training clips, 343 validation clips, and 387 testing clips. It covers 124 semantic categories, such as people, cars, and roads, with annotations surpassing those of previous video semantic segmentation (VSS) datasets. To validate the generalization of DTERN, we also conduct experiments on the Cityscapes dataset [6], which includes 2975 training clips, 500 validation clips, and 1275 testing clips covering 19 categories. Unlike VSPW, Cityscapes features higher resolution and is annotated every 30 frames, making it suitable for semi-supervised tasks.
Implementation details. Similar to Segformer [37], we use the MiT backbone pre-trained on ImageNet-1K as the encoder. Following prior works [31, 33], Our model adopts a $T = 4$ four-frame input configuration: for the target frame at time t, the input frame sequence is $\{t-9, t-6, t-3, t\}$. In the ADCB of the CAM, we set the kernel sizes as $\{k_x, k_y\} = 3$ and $k_g = 11$. For the PDCB, the kernel sizes are configured as $\{k_1, k_2, k_3, k_4\} = \{13, 11, 9, 7\}$ when the backbone adopts MiT-B0, otherwise, they are set to $\{k_1, k_2, k_3, k_4\} = \{11, 9, 7, 5\}$. In the VSPW experiments, the number of clusters for the MiT-B0, MiT-B1, MiT-B2, and MiT-B5 are set to 250, 200, 150, and 124. In the Cityscapes experiments, the number of clusters for the MiT-B0 and MiT-B1 are set to 60 and 45. All data preprocessing procedures and training strategy are consistent with

Methods	Type	Backbone	Params (M)↓	mIoU↑	Weighted IoU↑	VEC_8 ↑	VEC_{16} ↑	FPS↑	Sources
SegFormer [37]	Image	MiT-B0	3.8	33.1	57.1	30.1	31.3	96.1	NeurIPS (2021)
MRCFA [32]	Self-attention	MiT-B0	5.2	35.2	57.9	**33.4**	**34.5**	42.3	ECCV (2022)
CFFM [31]	Self-attention	MiT-B0	4.7	35.2	58.8	32.6	33.9	44.2	CVPR (2022)
CFFM++ [33]	other	MiT-B0	5.7	**35.9**	58.9	33.2	34.4	42.0	TPAMI (2024)
TFC [21]	other	MiT-B0	4.5	32.9	-	-	-	-	AAAI (2024)
Ours	-	MiT-B0	8.2	**35.9**	**59.8**	**33.4**	**34.5**	31.3	-
SegFormer [37]	Image	MiT-B1	13.8	37.7	59.5	33.8	35.0	88.2	NeurIPS (2021)
MRCFA [32]	Self-attention	MiT-B1	16.2	38.9	60.0	35.8	37.1	33.0	ECCV (2022)
CFFM [31]	Self-attention	MiT-B1	15.5	38.3	60.1	35.8	37.0	32.8	CVPR (2022)
CFFM++ [33]	other	MiT-B1	16.5	39.9	60.7	37.5	38.1	31.3	TPAMI (2024)
TFC [21]	other	MiT-B1	14.5	36.7	-	-	-	-	AAAI (2024)
TDSNet [39]	other	MiT-B1	15.4	39.2	60.2	-	-	-	Inf. Sci. (2025)
Ours	-	MiT-B1	18.1	**40.3**	**61.3**	**37.8**	**39.0**	27.2	-
PSPNet [44]	Image	ResNet-101	70.5	36.5	58.1	-	-	32.5	CVPR (2017)
OCRNet [40]	Image	ResNet-101	58.1	36.7	59.2	-	-	34.5	ECCV (2020)
Mask2Former [4]	Image	Swin-S	68.9	42.1	63.1	-	-	-	CVPR (2022)
MPVSS [35]	Optical-flow	Swin-S	108.0	40.4	62.0	-	-	-	NeurIPS (2023)
ETC [22]	Optical-flow	OCRNet	58.1	37.5	59.1	-	-	25.1	ECCV (2020)
NetWarp [36]	Optical-flow	OCRNet	58.1	37.5	58.9	-	-	6.2	ECCV (2018)
TCBst-ocr [24]	other	ResNet-101	58.1	37.4	59.3	-	-	8.1	CVPR (2021)
SegFormer [37]	Image	MiT-B2	24.2	44.1	63.7	40.3	41.5	54.2	NeurIPS (2021)
MRCFA [32]	Self-attention	MiT-B2	27.3	45.3	64.7	42.3	43.8	24.4	ECCV (2022)
CFFM [31]	Self-attention	MiT-B2	26.5	45.0	64.4	42.7	44.3	23.6	CVPR (2022)
CFFM++ [33]	other	MiT-B2	28.5	45.5	64.7	43.1	44.7	22.5	TPAMI (2024)
DCFM ($K = 1$) [45]	other	MiT-B2	24.8	45.0	64.3	-	-	-	TCSVT (2024)
TDSNet [39]	other	MiT-B2	26.5	45.9	65.1	-	-	-	Inf. Sci. (2025)
Ours	-	MiT-B2	29.1	**46.1**	**65.2**	**43.3**	**45.2**	20.6	-
SegFormer [37]	Image	MiT-B5	82.1	48.9	66.5	46.0	47.6	25.0	NeurIPS (2021)
MRCFA [32]	Self-attention	MiT-B5	84.5	49.9	66.0	45.9	46.8	11.4	ECCV (2022)
CFFM [31]	Self-attention	MiT-B5	85.5	49.3	66.5	47.1	48.2	10.6	CVPR (2022)
CFFM++ [33]	other	MiT-B5	87.5	50.1	66.5	47.6	48.7	10.4	TPAMI (2024)
TFC [21]	other	MiT-B5	82.8	48.3	-	-	-	-	AAAI (2024)
DCFM ($K = 1$) [45]	other	MiT-B5	82.1	49.5	66.0	-	-	-	TCSVT (2024)
Ours	-	MiT-B5	86.3	**50.3**	**66.7**	**48.1**	**49.3**	10.6	-

Table 1. Comparison with the state-of-the-art methods on the val set of VSPW [24]. The best results under each setting are highlighted in bold. The FPS is computed on a single NVIDIA GeForce RTX 4090.

those described in [31, 33]. The code is built on *mmsegmentation* [5] codebase.

Evaluation metrics. Following previous works [4, 21, 24, 31, 33, 39], we primarily employ the mean IoU and the weighted IoU as the main measures of segmentation performance. Previous works [21, 31, 33, 39] have adopted video consistency (VC) to evaluate the smoothness of the predicted segmentation maps. Formally, $\mathbf{VC}_C$ is computed as follows:

$$\mathrm{VC}_C = \frac{(\bigcap_{i=1}^{C} Q_{t+i}) \cap (\bigcap_{i=1}^{C} Q'_{t+i})}{(\bigcap_{i=1}^{C} Q_{t+i})} \tag{15}$$

where $\{\mathbf{Q}_{t+i}\}_{i=1}^{C}$ are the ground truth frames, $\{\mathbf{Q}'_{t+i}\}_{i=1}^{C}$ are the predicted labels.

We observe that the VC metric solely focuses on prediction consistency while neglecting the accuracy of segmentation and exhibiting insensitivity to small-object segmentation. For instance, a prediction of all zeros would result in a VC metric of 1, which is clearly undesirable. We introduce constraint 1: predictions must be effective:

$$\mathrm{VC}'_C = \frac{(\bigcap_{i=1}^{C} Q_{t+i} \cap \bigcap_{i=1}^{C} Q'_{t+i})}{(\bigcap_{i=1}^{C} Q_{t+i})} \tag{16}$$

The corrected VC′ metric considers both temporal consistency and segmentation accuracy. We further introduce constraint 2: we compute the mean IoU for consistent regions. Ultimately, our VEC metric is expressed as:

$$\mathrm{VEC}_C = \frac{1}{N_c} \sum_{k=1}^{N_c} \left(\frac{(\bigcap_{i=1}^{C} (Q_{t+i} = k) \cap \bigcap_{i=1}^{C} (Q'_{t+i} = k))}{(\bigcap_{i=1}^{C} (Q_{t+i} = k) \cup \bigcap_{i=1}^{C} (Q'_{t+i} = k))} \right) \tag{17}$$

where N_c denotes the category number. The VEC ensures both temporal consistency and segmentation effectiveness.

4.2. Segmentation Results

Performance on the VSPW: A comparison of DTERN with the state-of-the-art methods on the VSPW dataset is shown in Tab. 1. As in previous work [31, 33, 39], we also report the complexity of the models and categorize the models into two groups: small models (with less than 20M parameters) and large models (with more than 20M parameters). For small model, DTERN significantly outperforms the baseline SegFormer [37], achieving a 2.8% improvement in mIoU when using MiT-B0 as the backbone. DTERN also exhibits superior segmentation consistency, with VEC_8 and VEC_{16} increasing by 3.3% and 3.2%. When using the MiT-B1 as the backbone, DTERN achieves a 2.6% improvement in mIoU over SegFormer, with VEC_8 and VEC_{16} increasing by 4.0%. For larger models with MiT-B2 and MiT-B5 as the backbone, DTERN still significantly outperforms the baseline. DTERN provides consis-

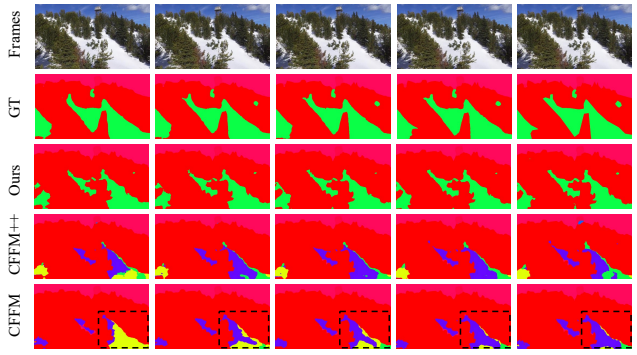


Figure 3. Qualitative results. From top to bottom: the input frames, the ground truth (GT), DTERN predictions, CFFM++ [33] predictions and CFFM [37] predictions.

tent performance improvement across all backbones, indicating the effectiveness of the proposed framework.

Performance on the Cityscapes: We also report the performance on the val set of the Cityscapes in Tab. 2. Similar to VSPW, DTERN also achieves better segmentation results than the baseline SegFormer[37]. When using MiT-B0 as the backbone, DTERN achieves a 2.6% improvement in mIoU over the baseline SegFormer, reaching 74.5% mIoU. With the deeper MiT-B1 as the backbone, the mIoU increases by 1.9%, achieving 76.0% mIoU.

Comparison with state-of-the-art methods: Under the same backbone, DTERN outperforms self-attention-based models such as CFFM [31] and CFFM++ [33] in terms of mIoU, weighted IoU, VEC_8, and VEC_{16}. For example, in the VSPW experiments, DTERN achieves improvements of 0.7%, 2.0%, 1.1% and 1.0% over CFFM when using MiT-B0, MiT-B1, MiT-B2, and MiT-B5 as backbones, respectively. Compared to CFFM++, DTERN achieves comparable mIoU results with MiT-B0 while demonstrating superior mIoU performance in all other cases. Additionally, DTERN achieves the best VEC_8 and VEC_{16} scores across all backbones. DTERN constructs local exemplars to model local temporal information, effectively enhancing the accuracy and stability of predictions. Moreover, by leveraging global exemplars to capture global temporal information, DTERN further improves the effective consistency of segmentation.

Qualitative results: We present qualitative examples of results in Fig. 3. We compare the predictions of DTERN with CFFM++ and CFFM on MiT-B1. It can be observed that the CFFM, which employs a reference-target interaction paradigm, produces unstable predictions even for adjacent frames where the same region (the black-box region) belongs to the same category. Although CFFM++ improves prediction consistency, consecutive prediction errors across frames. In contrast, DTERN demonstrates significantly improved segmentation accuracy and temporal consistency, delivering superior quantitative and qualitative results.

Methods	Type	Backbone	Params (M)↓	mIoU↑	Sources
FCN [23]	Image	MobileNetV2	9.8	61.5	CVPR (2015)
PSPNet [44]	Image	MobileNetV2	13.7	70.2	CVPR (2017)
DFF [47]	Optical-flow	ResNet-101	-	68.7	CVPR (2017)
GRFP [25]	Optical-flow	ResNet-101	-	69.4	ECCV (2018)
DVSN [38]	Optical-flow	ResNet-101	-	70.3	ECCV (2018)
Accel [17]	Optical-flow	ResNet-101	-	72.1	ECCV (2019)
ETC [22]	Optical-flow	ResNet-18	13.2	71.1	ECCV (2020)
SegFormer [37]	Image	MiT-B0	3.7	71.9	NeurIPS (2021)
MRCFA [32]	Self-attention	MiT-B0	4.2	72.8	ECCV (2022)
CFFM [31]	Self-attention	MiT-B0	4.6	74.0	CVPR (2022)
CFFM++ [33]	Self-attention	MiT-B0	5.1	74.3	TPAMI (2024)
Ours	-	MiT-B0	7.7	**74.5**	-
SegFormer [37]	Image	MiT-B1	13.8	74.1	NeurIPS (2021)
MRCFA [32]	Self-attention	MiT-B1	14.9	75.1	ECCV (2022)
CFFM [31]	Self-attention	MiT-B1	15.4	75.1	CVPR (2022)
CFFM++ [33]	other	MiT-B1	15.9	75.7	TPAMI (2024)
DCFM* [45]	other	MiT-B1	-	74.4	TCSVT (2024)
TDSNet [39]	other	MiT-B1	-	75.7	Inf. Sci. (2025)
Ours	-	MiT-B1	17.8	**76.0**	-

Table 2. Comparison with recent efficient VSS models on the Cityscapes val set [6]. The best results are highlighted in bold.

4.3. Ablation Study

For a fair comparison, we follow the training strategy in [31, 33] and conduct ablation studies with MiT-B1 on the VSPW dataset [24].

Effectiveness of GTEM: We conduct ablation experiments on GTEM to study its effect on the model. As shown in Tab. 3, the inclusion of the GTEM module results in a 0.2% improvement in both VEC_8 and VEC_{16}. This demonstrates that GTEM enhances the effective consistency of segmentation by capturing global temporal information through global exemplars. The slight decrease in mIoU can be attributed to the reference frame selection strategy, which enables the model to extract rich global temporal information relevant to the target frame. While GTEM further enhances segmentation consistency by exploiting global temporal information across the entire video, its stronger emphasis on global contexts may inadvertently dilute fine-grained details in the target frame features.

#	GTEM	SAM	CAM		mIoU↑	VEC_8 ↑	VEC_{16} ↑
			PDCB	ADCB			
a		✓	✓	✓	40.4	37.6	38.8
b	✓		✓	✓	40.0	37.1	38.0
c	✓	✓			39.7	36.7	37.8
d	✓	✓	✓		38.3	35.6	36.6
e	✓	✓		✓	39.6	37.1	38.1
f	✓	✓	✓	✓	40.3	37.8	39.0

Table 3. Ablation study of proposed blocks .

Effectiveness of CAM: To demonstrate the gain of CAM, we replace it with ResBlock with comparable parameters. As illustrated in Tab. 3, CAM achieves a 0.6% improvement in mIoU. This demonstrates that the proposed CAM enables more accurate cluster assignments compared to traditional stacked convolutional layers, leading to a significant improvement in segmentation performance. We also report ablation experiments on the dual-branch architecture of CAM. The results indicate that the removal of either the PDCB or ADCB leads to performance degradation. We ob-

#	Num	mIoU↑	VEC_8 ↑	VEC_{16} ↑
a	32	39.3	36.1	37.1
b	124	39.7	37.2	38.1
c	200	40.3	37.8	39.0
d	2048	39.6	37.0	37.9

Table 4. Ablation study on the number of cluster centers.

Method	Pearson↑		Spearman↑		Kendall↑	
	CFFM	CFFM++	CFFM	CFFM++	CFFM	CFFM++
Video Consistency	59.4	61.4	59.7	61.0	42.2	43.5
Video Effective Consistency	88.1	88.2	88.0	88.1	71.6	71.5

Table 5. Correlations between measured TC and ground-truth TC.

Methods	mIoU↑	VEC_8 ↑	VEC_{16} ↑	Sources
ClusterFormer [20]	38.0	34.9	36.3	NeurIPS (2023)
PaCaViT [10]	39.7	36.8	38.5	CVPR (2023)
Ours	40.3	37.8	39.0	-

Table 6. Ablation study on clustering strategy.

serve a sharp drop in mIoU when the ADCB is removed, which indicates the benefits of non-local features. When both branches are incorporated, DTERN achieves optimal performance with a 40.3% mIoU. This highlights the synergistic improvement between the two branches. Furthermore, we provide a visualization of our cluster assignments generated by CAM in Fig. 4.

Effectiveness of SAM: To validate the benefits of SAM, we substitute our cluster center aggregation strategy with direct summation across frames. The experimental results in Tab. 3 demonstrate that this additive operation causes a 0.3% mIoU degradation. Complex variations between video frames may cause spatial discrepancies in cluster centers and lead to unstable predictions, which SAM effectively mitigates, resulting in improved model performance.

Influence of the number of cluster centers: Table 4 explores the impact of the cluster-category cardinality relationship (K vs. C) on segmentation performance: the optimal mIoU is achieved when K is slightly larger than C (e.g., $K = 200$); $K < C$ (e.g., $K = 32$) limits feature discriminability, leading to inter-class confusion (with a 1% drop in mIoU); when $K = C$, rigid correspondences reduce performance in complex scenes (with a 0.6% drop in mIoU); and $K >> C$ (e.g., $K = 2048$) introduces computational redundancy and overfitting (with a 0.7% drop in mIoU). The results indicate that a moderate number of clusters ($K > C$) can generate finer-grained feature representations for complex scenes, enhancing flexibility and expressive power.

Effectiveness of VEC: To validate the effectiveness of the VEC, we conduct experiments based on the *Measuring Temporal Consistency* [43] to evaluate VC and VEC. As shown in Tab. 5, in terms of Pearson, Spearman, and Kendall correlation coefficients, VEC exhibits significantly stronger correlation with the ground-truth TC compared to the video consistency measure. We also demonstrate the

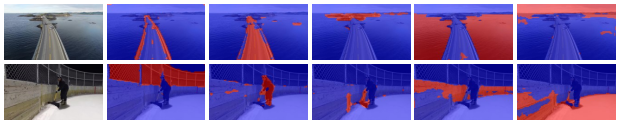


Figure 4. Qualitative results of our cluster assignments. Different cluster assignments effectively focus on different category regions. For certain categories, we generate diverse cluster assignments.

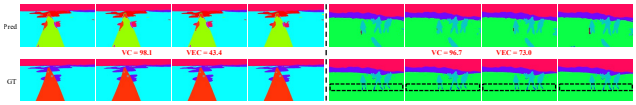


Figure 5. Qualitative comparison of VC and VEC metrics using CFFM++ predictions.

qualitative analysis of the VC and VEC in Fig. 5.

Effectiveness of clustering strategy: In Tab. 6, we also compare our clustering strategy with recently adopted clustering strategies [10, 20] in image segmentation. Our strategy outperforms the others. The efficacy of our approach stems from its capability to address two inherent challenges: 1) complex variations in illumination, object morphology, and other scene dynamics that render traditional stacked convolutions inadequate for precise cluster assignments, and 2) prediction instability caused by cluster assignment discrepancies between adjacent LTEMs and inter-frame cluster center variations. Our CAM improves cluster assignment accuracy and SAM mitigates instability, ultimately enhancing segmentation performance.

5. Conclusion

For video semantic segmentation, the effective utilization of both local and global temporal information is crucial. In this paper, the primary contributions are two folds: 1) a novel Video Effective Consistency metric for evaluating temporal consistency and segmentation effectiveness simultaneously and 2) the Dual-Temporal Exemplar Representation Network (DTERN), leveraging the strong representational capability of cluster centers to effectively model local and global temporal information and achieving more accurate and efficient segmentation results. To the best of our knowledge, this is the first work to implement end-to-end video semantic segmentation by utilizing exemplars. DTERN introduces two key modules: the Local Temporal Exemplar Module and the Global Temporal Exemplar Module. The former constructs local exemplars to model local temporal information, enabling stable prediction of the target frame, thereby enhancing segmentation accuracy. The latter leverages global exemplars to dynamically learn global temporal information, improving the effective consistency of segmentation. With a limited increase in parameters, DTERN achieves improved segmentation performance.

6. Acknowledgments

This work was supported by the National Science Fund for Distinguished Young Scholars, PR China under Grant No. 62025601, Transformation Foundation of Tianfu Jincheng Laboratory (2025ZH013) and National Natural Science Foundation of China under Grant 62376174.

References

[1] Joao Carreira, Viorica Patraucean, Laurent Mazare, Andrew Zisserman, and Simon Osindero. Massively parallel video networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 649–666, 2018. 2

[2] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 2

[3] Wanli Chen, Xinge Zhu, Ruoqi Sun, Junjun He, Ruiyu Li, Xiaoyong Shen, and Bei Yu. Tensor low-rank reconstruction for semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, pages 52–69. Springer, 2020. 2

[4] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 6

[5] MMSegmentation Contributors. Mmsegmentation: Openmmlab semantic segmentation toolbox and benchmark, 2020. 6

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 5, 7

[7] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015. 1

[8] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3146–3154, 2019. 2

[9] Raghudeep Gadde, Varun Jampani, and Peter V Gehler. Semantic video cnns through representation warping. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4453–4462, 2017. 1, 2

[10] Ryan Grainger, Thomas Paniagua, Xi Song, Naresh Cuntoor, Mun Wai Lee, and Tianfu Wu. Paca-vit: Learning patch-to-cluster attention in vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 8

[11] Diandian Guo, Deng-Ping Fan, Tongyu Lu, Christos Sakaridis, and Luc Van Gool. Vanishing-point-guided video semantic segmentation of driving scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3544–3553, 2024. 2

[12] Chi-Wei Hsiao, Cheng Sun, Hwann-Tzong Chen, and Min Sun. Specialize and fuse: Pyramidal output representation for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7137–7146, 2021. 2

[13] Hanzhe Hu, Deyi Ji, Weihao Gan, Shuai Bai, Wei Wu, and Junjie Yan. Class-wise dynamic graph convolution for semantic segmentation. In *European Conference on Computer Vision*, pages 1–17. Springer, 2020. 2

[14] Po-Yu Huang, Wan-Ting Hsu, Chun-Yueh Chiu, Ting-Fan Wu, and Min Sun. Efficient uncertainty estimation for semantic segmentation in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 520–535, 2018. 1

[15] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 603–612, 2019. 2

[16] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017. 1

[17] Samvit Jain, Xin Wang, and Joseph E Gonzalez. Accel: A corrective fusion network for efficient semantic segmentation on video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8866–8875, 2019. 2, 7

[18] Jiangtong Li, Wentao Wang, Junjie Chen, Li Niu, Jianlou Si, Chen Qian, and Liqing Zhang. Video semantic segmentation via sparse temporal transformer. In *Proceedings of the 29th ACM International Conference on Multimedia*, page 59–68, New York, NY, USA, 2021. Association for Computing Machinery. 1, 2

[19] Yule Li, Jianping Shi, and Dahua Lin. Low-latency video semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5997–6005, 2018. 2

[20] James C Liang, Yiming Cui, Qifan Wang, Tong Geng, Wenguan Wang, and Dongfang Liu. Clusterformer: Clustering as a universal visual learner. In *Neural Information Processing Systems (NeurIPS)*, 2023. 8

[21] Matthieu Lin, Jenny Sheng, Yubin Hu, Yangguang Li, Lu Qi, Andrew Zhao, Gao Huang, and Yong-Jin Liu. Exploring temporal feature correlation for efficient and stable video semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3450–3458, 2024. 2, 6

[22] Yifan Liu, Chunhua Shen, Changqian Yu, and Jingdong Wang. Efficient semantic video segmentation with per-frame

inference. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 352–368. Springer, 2020. 2, 6, 7

[23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 2, 7

[24] Jiaxu Miao, Yunchao Wei, Yu Wu, Chen Liang, Guangrui Li, and Yi Yang. Vspw: A large-scale dataset for video scene parsing in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4133–4143, 2021. 2, 5, 6, 7

[25] David Nilsson and Cristian Sminchisescu. Semantic video segmentation by gated recurrent flow propagation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6819–6828, 2018. 1, 2, 7

[26] Yuval Nirkin, Lior Wolf, and Tal Hassner. Hyperseg: Patch-wise hypernetwork for real-time semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4061–4070, 2021. 2

[27] Matthieu Paul, Martin Danelljan, Luc Van Gool, and Radu Timofte. Local memory attention for fast video semantic segmentation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1102–1109, 2021. 2

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 2

[29] Evan Shelhamer, Kate Rakelly, Judy Hoffman, and Trevor Darrell. Clockwork convnets for video semantic segmentation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pages 852–868. Springer, 2016. 2

[30] Baifeng Shi, Siyu Gai, Trevor Darrell, and Xin Wang. Toast: Transfer learning via attention steering. *arXiv preprint arXiv:2305.15542*, 2023. 4

[31] Guolei Sun, Yun Liu, Henghui Ding, Thomas Probst, and Luc Van Gool. Coarse-to-fine feature mining for video semantic segmentation. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3126–3137, 2022. 1, 2, 5, 6, 7

[32] Guolei Sun, Yun Liu, Hao Tang, Ajad Chhatkuli, Le Zhang, and Luc Van Gool. Mining relations among cross-frame affinities for video semantic segmentation. In *European Conference on Computer Vision*, pages 522–539. Springer, 2022. 6, 7

[33] Guolei Sun, Yun Liu, Henghui Ding, Min Wu, and Luc Van Gool. Learning local and global temporal contexts for video semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 2, 3, 5, 6, 7

[34] Hao Wang, Weining Wang, and Jing Liu. Temporal memory attention for video semantic segmentation. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2254–2258. IEEE, 2021. 1, 2

[35] Yuetian Weng, Mingfei Han, Haoyu He, Mingjie Li, Lina Yao, Xiaojun Chang, and Bohan Zhuang. Mask propagation for efficient video semantic segmentation. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 6

[36] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018. 6

[37] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34: 12077–12090, 2021. 5, 6, 7

[38] Yu-Syuan Xu, Tsu-Jui Fu, Hsuan-Kung Yang, and Chun-Yi Lee. Dynamic video segmentation network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6556–6565, 2018. 2, 7

[39] Haochen Yuan, Junjie Peng, and Zesu Cai. Tdsnet: A temporal difference based network for video semantic segmentation. *Information Sciences*, 686:121335, 2025. 6, 7

[40] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pages 173–190. Springer, 2020. 6

[41] Yichen Yuan, Lijun Wang, and Yifan Wang. Csanet for video semantic segmentation with inter-frame mutual learning. *IEEE Signal Processing Letters*, 28:1675–1679, 2021. 1

[42] Fan Zhang, Yanqin Chen, Zhihang Li, Zhibin Hong, Jingtuo Liu, Feifei Ma, Junyu Han, and Errui Ding. Acfnet: Attentional class feature network for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6798–6807, 2019. 2

[43] Yizhe Zhang, Shubhankar Borse, Hong Cai, Ying Wang, Ning Bi, Xiaoyun Jiang, and Fatih Porikli. Perceptual consistency in video segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2564–2573, 2022. 8

[44] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 6, 7

[45] Yaoyan Zheng, Hongyu Yang, and Di Huang. Deep common feature mining for efficient video semantic segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(12):12991–13003, 2024. 6, 7

[46] Zilong Zhong, Zhong Qiu Lin, Rene Bidart, Xiaodan Hu, Ibrahim Ben Daya, Zhifeng Li, Wei-Shi Zheng, Jonathan Li, and Alexander Wong. Squeeze-and-attention networks for semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13065–13074, 2020. 2

[47] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2349–2358, 2017. 1, 2, 7

[48] Zhen Zhu, Mengde Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 593–602, 2019. 2