Supplementary for Video Semantic Segmentation via Sparse Temporal Transformer

Jiangtong Li*
MoE Key Lab of Artificial Intelligence,
Shanghai Jiao Tong University
keep_moving-Lee@sjtu.edu.cn

Wentao Wang*
MoE Key Lab of Artificial Intelligence,
Shanghai Jiao Tong University
wwt117@sjtu.edu.cn

Junjie Chen
MoE Key Lab of Artificial Intelligence,
Shanghai Jiao Tong University
chen.bys@sjtu.edu.cn

Li Niu†
MoE Key Lab of Artificial Intelligence,
Shanghai Jiao Tong University
ustcnewly@sjtu.edu.cn

Jianlou Si
SenseTime Research,
SenseTime
sijianlou@sensetime.com

Chen Qian
SenseTime Research,
SenseTime
qianchen@sensetime.com

Liqing Zhang†
MoE Key Lab of Artificial Intelligence,
Shanghai Jiao Tong University
zhang-lq@cs.sjtu.edu.cn

CCS CONCEPTS

• **Computing methodologies** → **Video segmentation**.

KEYWORDS

semantic segmentation, video semantic segmentation, transformer, temporal consistency, semi-supervised learning

ACM Reference Format:
Jiangtong Li, Wentao Wang, Junjie Chen, Li Niu, Jianlou Si, Chen Qian, and Liqing Zhang. 2021. Supplementary for Video Semantic Segmentation via Sparse Temporal Transformer. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3474085.3475409

In this document, we provide additional materials to supplement our main submission. In Sec. 1, we introduce some detailed information about our experiment setting. In Sec. 2, we study the effect of different numbers of key frames. In Sec. 3, we visualize the query selection results. In Sec. 4, we visualize the attention map between the query feature with its corresponding key features. In Sec. 5, we conduct additional qualitative comparisons to show the superiority of our proposed Sparse Temporal Transformer (STT) method. In Sec. 6, we attach a video file "video example.mp4" in the supplementary to compare our method with two state-of-the-art methods.

*Both authors contributed equally to this research.
†Corresponding Authors.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MM '21, October 20–24, 2021, Virtual Event, China
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475409

1 EXPERIMENT SETTING

1.1 Dataset

1.1.1 Cityscapes. is collected for urban scene understanding and contains 30-frame snippets of the street scene with 17 frames per second. The dataset contains 5,000 high quality pixel-level finely annotated images at 20th frame in each snippets with 19 semantic categories, which are divided into 2,975, 500 and 1,525 images for training, validation and testing, respectively.

1.1.2 Camvid. is an automotive dataset. It contains five different videos with 11 semantic categories, which has ground truth labels every 30 frames. The annotated frames are grouped into 367, 100, and 233 for training, validation and testing, respectively.

1.2 Speed Measurement

All testing experiments are conducted with the batch size 1 on a single Nvidia Tesla Volta 100 GPU in the PyTorch framework. We find that previous methods are implemented with different deep-learning frameworks and evaluated on different types of devices, so for consistent comparisons, we follow the same strategy as TD-Net [4] and report the speed of these previous methods based on benchmark-based conversions and our re-implementations.

1.3 Implementation Details

On both datasets, we implement all the methods on PyTorch [6] and train them on 4 Nvidia Tesla Volta 100 GPUs with cross-entropy as the loss function. The segmentation networks in this paper are trained by minibatch stochastic gradient descent (SGD) [1] with momentum 0.9, weight decay 5e-4 and batch size 8 for 80k iterations. For the first 40k iterations, we only train our method with labeled data, and for the rest 40k iteration, we train our method on both labeled and unlabeled data. The learning rate is initialized as 0.01 and is multiplied by $(1 - \frac{iter}{maxiter})^{0.9}$. Normal data augmentation methods are applied during training, such as random scaling (from 0.5 to 2.0) and random flipping. During testing, we resize the output

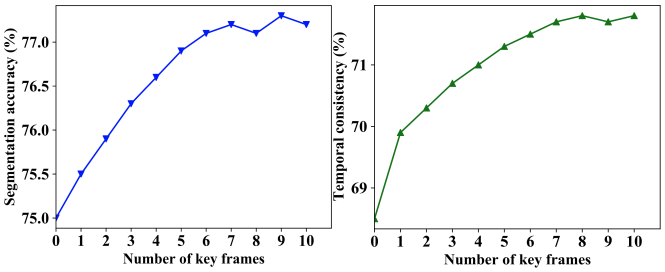

Figure 1: The effect of using different numbers of key frames. The left (*resp.*, right) sub-figure shows how segmentation accuracy (*resp.*, temporal consistency) changes along with the number of key frames change. When the number of key frames equals to zeros, we only use the PSPNet18 without our STT method.

to the input's original resolution for evaluation. Besides, for the temporal transformer, we recall seven previous frames as the key frames to interact with the current frame. As for the query selection, the neighboring ratio r is set as 5. As for the key selection, the start size s, end size e and the expansion coefficient ϵ are set as 1, 4 and 1, respectively. Therefore, for each query feature, the corresponding key size will be $3^2+5^2+7^2+9^2+9^2+9^2+9^2 = 407$. On Cityscapes [3] (*resp.*, Camvid [2]), we randomly cut the images into 768×768 (*resp.*, 640×640) as the training input. All the hyper-parameters are decided based on the validation results.

2 EFFECT OF THE NUMBER OF KEY FRAMES

In Figure 1, we show the effect of using different numbers of key frames. We can find that when the number of key frames changes from 0 to 1, the improvement in both segmentation accuracy and temporal consistency are the largest. As the number of key frames increases, the improvement of adding another extra key frame is getting smaller. Besides, when the number of key frames is larger than 7, the segmentation accuracy and temporal consistency get stable, which indicates that sufficient temporal relations have been captured by our STT method.

3 QUERY SELECTION VISUALIZATION

We visualize the query selection in Figure 3. The RGB frames and their corresponding semantic segmentation labels are shown in the first row and the second row respectively. The unselected regions in the RGB frames and the segmentation labels are covered with white masks. From these two examples, we can find that our proposed NSM is fully capable of distinguishing the complex regions from the simple ones. Apart from that, we also observe that the selected query regions are mainly the semantic boundary instead of the object boundary or color boundary. Considering that query selection is performed based on the feature map from the encoder, this also indicates that the encoder could capture the semantic information among visual objects.

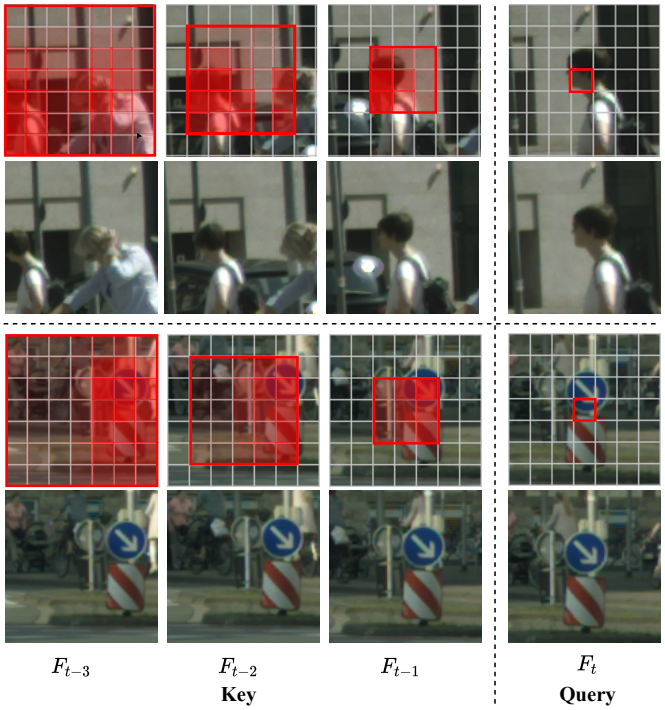

Figure 2: The visualization of attention map between the query point and the key regions. The red boxes in "Query" (*resp.*, "Key") enclose the the query point (*resp.*, key regions). The attention similarity between the query feature and key regions is represented by color, where deeper color indicates higher similarity. Best viewed by zooming in.

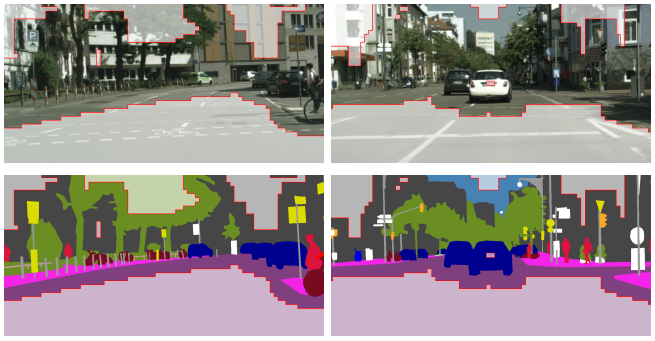

Figure 3: The visualization of the query selection results. The unselected regions in the RGB frames and the segmentation labels are covered with white mask and the boundary between selected and unselected regions are marked with red lines. Best viewed by zooming in.

4 ATTENTION VISUALIZATION

In Figure 2, we show the attention map between the query feature and its corresponding key regions based on the similarities between them (see Equ. 6 in the main submission), where deeper color indicates higher similarity. These two examples show that the

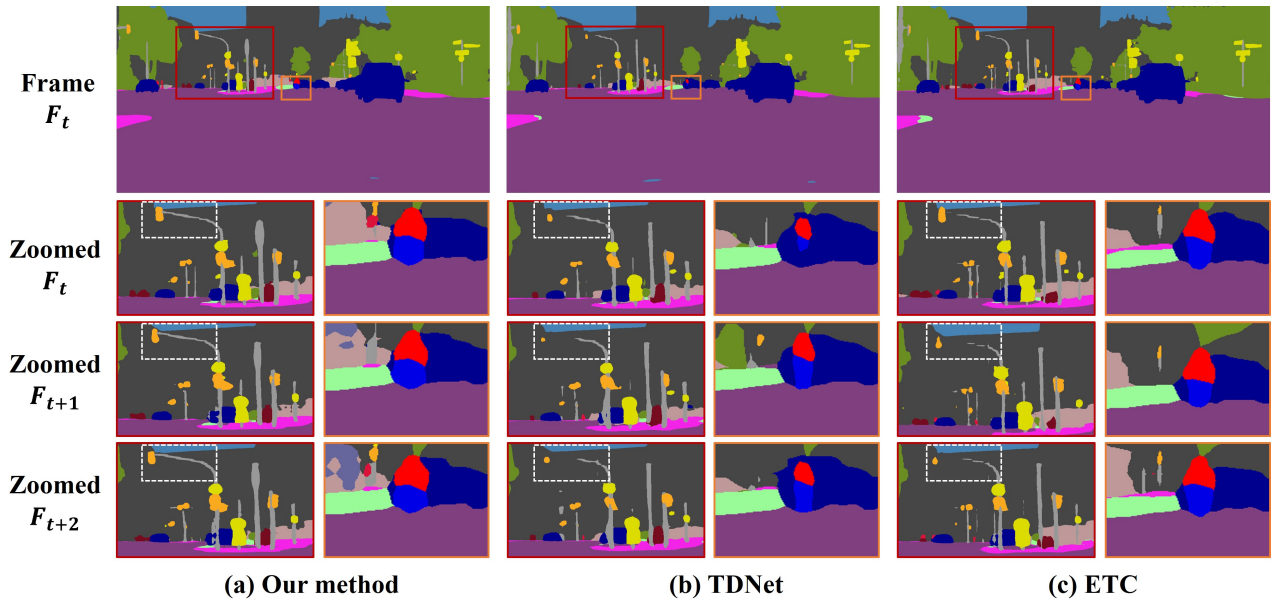

Figure 4: More segmentation results of our method compared with other two methods, *i.e.*, TDNet [4] and ETC [5]. In red box, streetlight generated by our model is complete while discontinuous in the other methods. In orange box, our model and ETC are able to generate the motorcyclist (the one in red and blue) reasonably while TDNet fails.

multi-head attention module in the temporal transformer is capable of capturing the temporal relation according to the selected query points, like the boy in the first example and the sign in the second example. Besides, in the first example, we can find that given the the boy as query, our temporal transformer not only captures himself in the key region, but also captures the old woman in the searching regions. Since the boy and the woman both belong to "human", this result shows that our model could not only capture the temporal relation of the same object, but also capture the temporal relation from other semantically similar objects.

5 QUALITATIVE COMPARISONS

In Figure 4, Figure 5 and Figure 6, we show more qualitative results compared with two representative baselines (TDNet [4] and ETC [5]). Following TDNet [4] and ETC [5], we employ all three segmentation model based on PSPNet18 [7] and all experiments are conducted on Cityscapes [3]. On the top part of Figure 4, Figure 5 and Figure 6, we show full-size segmentation results of frame F_t. For better visualization, we zoom the region in the red and orange box across 3 frames in the bottom part of figure. It can be seen that our model can generate more complete objects with continuous structure (*e.g.*, red box in Figure 4 and Figure 5, orange box in Figure 6). Similarly, the results generated by our model are more reasonable (*e.g.*, red box in Figure 6, orange box in Figure 4 and Figure 5) compared to other methods.

6 VIDEO EXAMPLE

In the video example, we randomly select 10 video clips (30 frames in 1 video clip) from Cityscapes validation set and use it to generate the corresponding semantic labels by means of our STT method, TDNet [4] and ETC [5]. In the video, the original RGB video is on the left top (Row 1, Col 1) and our STT method is on the right top (Row 1, Col 2). The TDNet is on the left bottom (Row 2, Col 1) and ETC is on the right bottom (Row 2, Col 2). Compared to TDNet and ETC, our method is able to generate more reasonable objects with continuous boundaries. Besides, our method is able to generate more stable and temporally consistent video results with less noises.

REFERENCES

[1] Léon Bottou, Frank E Curtis, and Jorge Nocedal. 2018. Optimization methods for large-scale machine learning. *Siam Review* 60, 2 (2018), 223–311.
[2] Gabriel J Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. 2008. Segmentation and recognition using structure from motion point clouds. In *ECCV 2008*.
[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The cityscapes dataset for semantic urban scene understanding. In *CVPR 2016*.
[4] Ping Hu, Fabian Caba, Oliver Wang, Zhe Lin, Stan Sclaroff, and Federico Perazzi. 2020. Temporally distributed networks for fast video semantic segmentation. In *CVPR 2020*.
[5] Yifan Liu, Chunhua Shen, Changqian Yu, and Jingdong Wang. 2020. Efficient Semantic Video Segmentation with Per-frame Inference. In *ECCV 2020*.
[6] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NeurIPS Autodiff Workshop 2017*.
[7] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2017. Pyramid scene parsing network. In *CVPR 2017*.

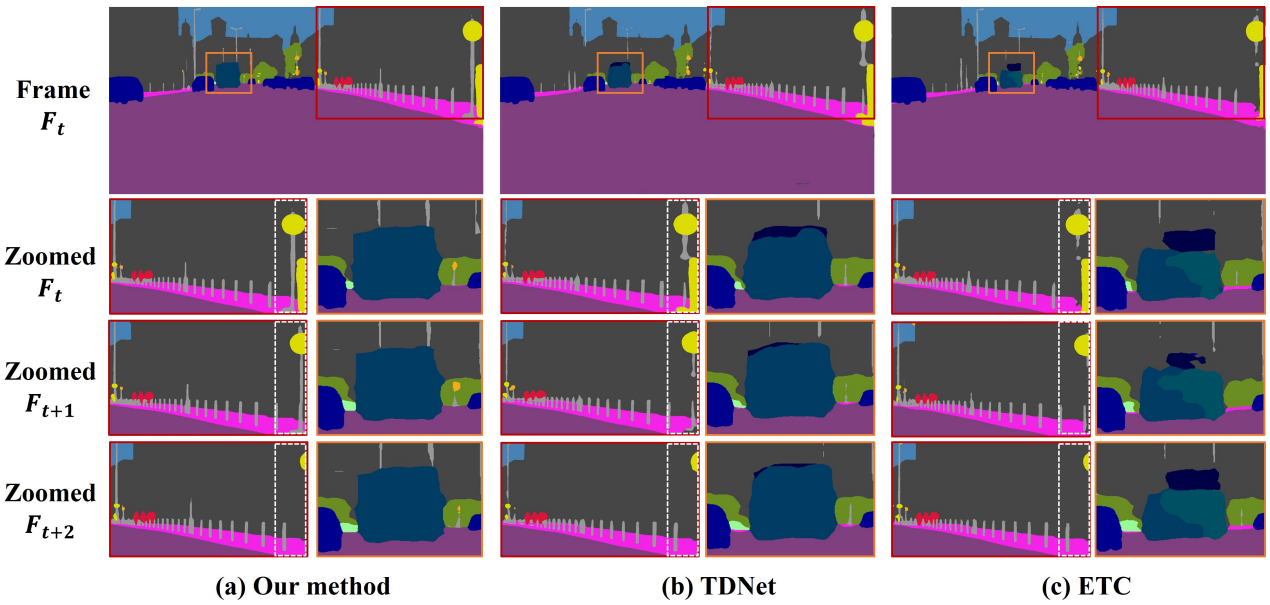

Figure 5: More segmentation results of our method compared with other two methods, *i.e.*, TDNet [4] and ETC [5]. In red box, the sign generated by our model is complete while discontinuous in the other methods. In orange box, our model is able to generate the bus (the one in dark green) more reasonable compared to the other methods.

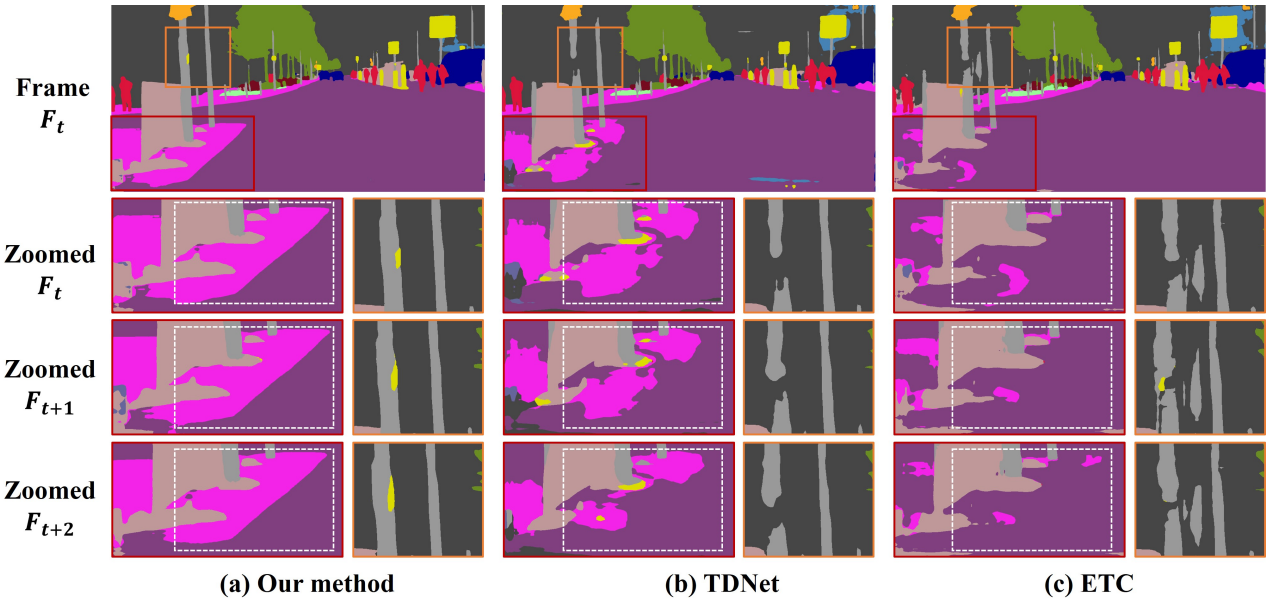

Figure 6: More segmentation results of our method compared with other two methods, *i.e.*, TDNet [4] and ETC [5]. In red box, the sidewalk (the pink area) generated by our model is complete while discontinuous or disappeared in the other methods. In orange box, our model is able to generate lamp pole continuous while discontinuous in the other methods.