DEFORMABLE VISTR: SPATIO TEMPORAL DEFORMABLE ATTENTION FOR VIDEO INSTANCE SEGMENTATION

Sudhir Yarram[†], *Jialian Wu*[†], *Pan Ji*[‡], *Yi Xu*[‡], *Junsong Yuan*[†]

[†] University at Buffalo [‡]OPPO US Research Center, InnoPeak Technology Inc.

ABSTRACT

Video instance segmentation (VIS) task requires classifying, segmenting, and tracking object instances over all frames in a video clip. Recently, VisTR [1] has been proposed as end-to-end transformer-based VIS framework, while demonstrating state-of-the-art performance. However, VisTR is slow to converge during training, requiring around 1000 GPU hours due to the high computational cost of its transformer attention module. To improve the training efficiency, we propose Deformable VisTR, leveraging spatio-temporal deformable attention module that only attends to a small fixed set of key spatio-temporal sampling points around a reference point. This enables Deformable VisTR to achieve linear computation in the size of spatio-temporal feature maps. Moreover, it can achieve on par performance as the original VisTR with 10× less GPU training hours. We validate the effectiveness of our method on the Youtube-VIS benchmark. Code is available at https://github.com/skrya/DefVIS.

***Index Terms*—** video instance segmentation; deformable convolution; efficient framework.

1. INTRODUCTION

Video instance segmentation task [2] requires classifying, segmenting instances in each frame, and tracking the same instance across frames. It is more challenging as we need to perform instance segmentation for each individual frame and also establish data association of instances across consecutive frames *a.k.a.*, tracking. Different from previous methods that rely on sophisticated pipelines [3, 2, 4], the recent work VisTR [1] was proposed as an end-end trainable framework to achieve state-of-the-art results. Despite the interesting design and good performance, VisTR requires much longer training time (epochs) to converge. For example, on Youtube-VIS benchmark [2], VisTR needs around 1000 GPU hours (500 training epochs[1]) to converge on a NVIDIA Tesla V100 GPU. This issue mainly arises due to the shortcoming of transformer attention modules to process spatio-temporal clip level features. During initialization, attention weights are uniformly distributed to all the pixels in the feature maps and long training hours is necessary for attention weights to be learned to focus on specific pixels. Also, the computational complexity of attention weights computation in Transformer encoder is of $O(H^2W^2T^2C)$, where H, W, T, C corresponds to the height, weight, temporal span, and channel dimension of the features belonging to the input clip, respectively. Therefore, it requires quadratic computation with respect to pixel numbers in the input clip (Table 1). Thus, it is of very high computational complexity to process a video clip.

In the image domain, deformable convolution [5] can be seen as self-attention mechanism that can attend to sparse meaningful locations. Leveraging it, Deformable DETR [6] mitigates the slow convergence and high complexity of DETR [7], a transformer framework for object detection. Since, VisTR is inspired from DETR, it is natural that deformable attention can also be extended to address the issues of VisTR. However, Defomable DETR was proposed for image domain, it is not directly applicable to video domain which needs to address both spatial and temporal dimension.

In this paper, we propose Deformable VisTR, which alleviates the high complexity and slow convergence issues of VisTR. It combines the spatio-temporal sparse sampling of deformable convolution and relation modeling capability of transformers to achieve linear computation with respect to pixel numbers. This brings significant improvement from quadratic complexity of VisTR.

To implement Deformable VisTR, we propose a spatio-temporal deformable attention module, which attends to small set of key sampling locations (say K, where $K << HWT$) out of all the spatio-temporal feature map pixels (HWT) for each reference point. Moreover, these prominent key sampling points are predicted from the feature at the reference point. By replacing the attention modules used in the transformer encoder and decoder of VisTR with spatio-temporal deformable attention module, the computation complexity can be reduced to $O(HWTCK)$, thus achieving linear computation in spatio-temporal feature map size of an input clip (see Table 1).

We evaluate the effectiveness of our method on Youtube-VIS benchmark. Compared with VisTR, Deformable VisTR can achieve on par performance with 10× less GPU training hours.

[1] The epoch number is 18 in VisTR codebase but actually equals to ∼500 epochs in common practice, as it trains each video ∼28 times in one epoch.

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

Method	Comp. Complexity	Training time (GPU Hours)	Training Epochs	Accuracy (mAP(%))
VisTR [1]	$O(H^2W^2T^2C)$	1000	∼500	35.6
Deformable VisTR	$O(HWTCK)$	**120**	**50**	34.6

Table 1. **Training convergence comparision of VisTR and Deformable VisTR using ResNet-50 backbone on Youtube-VIS val set.** H, W, T, C corresponds to the feature map height, weight, temporal span and channel dimension for the input video clip, respectively. Each reference point attends to a set of K locations out of all spatio-temporal (HWT) locations. GPU hours are evaluated on a NVIDIA Tesla V100 GPU. Our proposed Deformable VisTR can achieve on par performance with 10× less training epochs and time in GPU hours.

2. RELATED WORK

2.1. Video Instance Segmentation

The VIS task [2] has received lot of attention recently. Several state-of-the-art methods [2, 3, 4] typically develop sophisticated pipelines to tackle it. Top-down approaches [3, 2] adopt the tracking-by-detection paradigm, depending mainly on image-level instance segmentation models [8, 9] and complex instance association rules designed by human. Bottom-up approaches [4] differentiate object instances by clustering learned pixel embeddings. These methods need to employ multiple iterations to generate the masks due to heavy reliance on the dense prediction quality, which makes them slow. In contrast, VisTR proposes to build a simple and end-to-end trainable VIS framework. However, VisTR suffers from long training time which we address by leveraging the idea of deformable convolution. We try to build an end-to-end VIS framework, which is efficient and converges fast. For additional implementation details of VisTR, please refer to the manuscript [1].

2.2. Deformable Attention Mechanism

As discussed in [10], deformable convolution [5], a variant of convolution can be seen as self-attention mechanism. Particularly, deformable convolution is shown to operate more efficiently on image recognition than transformer self-attention. Deformable DETR [6] proposes deformable attention where each query element focuses on a fixed set of key sampling points predicted from the features of query element. However, deformable attention was proposed for image domain. We extend it to video domain by proposing spatio-temporal deformable attention module.

3. PROPOSED METHOD

3.1. Revisiting Multi-Head Attention in Transformer

Attention mechanisms is the core component of Transformer network architecture [11]. Given a query element (*e.g.,* a target feature corresponding to a 2d pixel position in input image) and a set of key elements (*e.g.,* features belonging to other pixel positions in the same input image), the multi-head-attention module adaptively aggregates the key contents according to the attention weights that measure the agreement of query-key pairs. Different attention heads are used for the model to focus on the contents from different representation subspaces and different positions. Let Ω_q specify the set of query elements and $q \in \Omega_q$ indexes a query element with representation feature $z_q \in \mathbb{R}^C$, where C is the feature dimension. Let Ω_k specify the set of key elements and $k \in \Omega_k$ indexes a key element with representation feature $x_k \in \mathbb{R}^C$. To differentiate spatial positions, the representation features z_q and x_k are usually of summation of the element contents and positional embeddings. Then the multi-head attention can be formulated as follows

$$MultiHeadAttn(\boldsymbol{z}_q, \boldsymbol{x}) = \sum_{m=1}^{M} \boldsymbol{W}_m \big[\sum_{k \in \Omega_k} A_{mqk} \cdot \boldsymbol{W}^{'}_m \boldsymbol{x}_k \big], \quad (1)$$

where m indexes the attention head and M is the total number of attention heads. $\boldsymbol{W}^{'}_m \in \mathbb{R}^{C_v \times C}$ and $\boldsymbol{W}_m \in \mathbb{R}^{C \times C_v}$ are learnable weights. $C_v = C/M$. The attention weights are normalized as $\sum_{k \in \Omega_k} A_{mqk} = 1$, where $A_{mqk} \propto exp\{\frac{\boldsymbol{z}_q^T \boldsymbol{U}_m^T \boldsymbol{V}_m \boldsymbol{x}_k}{\sqrt{C_v}}\}$ and $\boldsymbol{U}_m, \boldsymbol{V}_m \in \mathbb{R}^{C_v \times C}$ are learnable weights.

While multi head attention modules have shown to be effective, they still need long training schedules before convergence. Suppose the number of query and key elements are of N_q and N_k, respectively. At initialization, $\boldsymbol{U}_m \boldsymbol{z}_q$ and $\boldsymbol{V}_m \boldsymbol{x}_k$ follow normal distribution with mean of 0 and variance of 1, which makes attention weights $A_{mqk} \approx \frac{1}{N_k}$, when N_K is large. During backpropogation, it will lead to uncertain gradient computation for input features. Thus, long training hours is necessary for attention weights to be learned to focus on specific keys. In the video domain, where the key elements are usually clip-level spatio-temporal pixels, N_k can be very large and convergence is slow.

Moreover, the computational and memory complexities for the multi-head attention can be very high. For Eq 1, the computational complexity is $O(N_qC^2 + N_kC^2 + N_qN_kC)$. In the video domain, where the query and key elements are both of video pixels, $N_q = N_k >> C$, the complexity is dominated by the third term, as $O(N_qN_kC)$.

VisTR exploits a standard Transformer encoder-decoder architecture to process input video clip whose features obtained from backbone $\in \mathbb{R}^{C \times T \times H \times W}$, where H, W, T denote the feature map height, width, and temporal-span respectively. For the transformer encoder, $N_q = N_k = HWT$, So, the computational complexity is $O(H^2W^2T^2C)$. For the transformer decoder, $N_q = N$, where N is the number of object queries (*e.g.,* 360) and $N_k = HWT$. The computational complexity is $O(NHWTC)$. Thus, the multi-head attention module suffers from a quadratic complexity growth with the

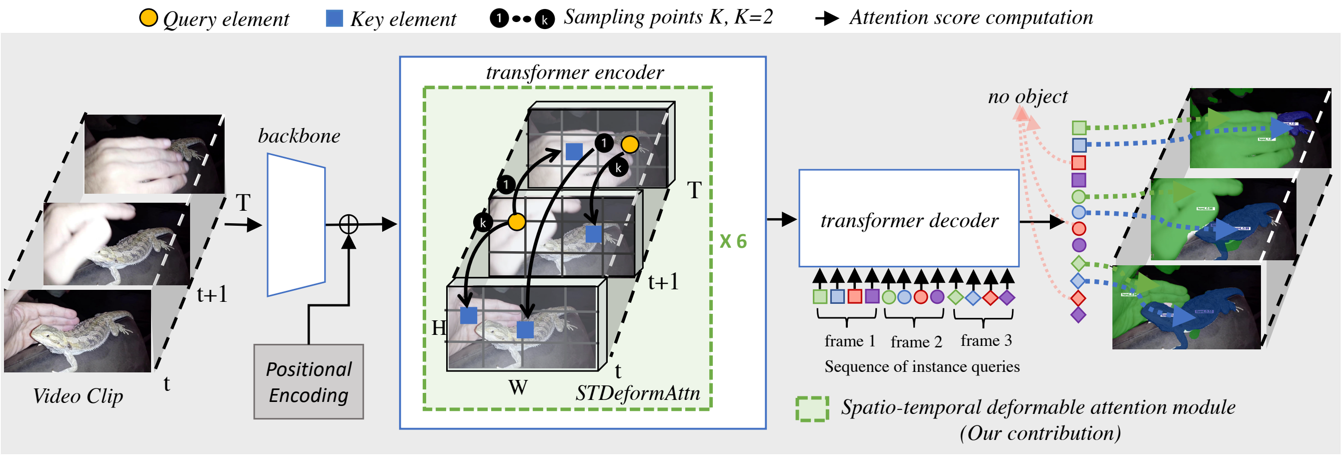

Fig. 1. **Illustration of proposed Deformable VisTR.** The green dashed box demonstrates our contribution, spatio-temporal deformable attention (STDeformAttn) module. In transformer encoder of VisTR [1], for each of the query element (●), all possible features in spatio-temporal feature map (HWT) are used as key elements in attention score computation. However, in deformable VisTR, for each query element (●), only a small fixed set of sampling points ($K, K << HWT$) predicted from query feature are used as key elements (■) during attention. In this illustration, $K = 3$. Though we do not illustrate, the STDeformAttn module is also used in the transformer decoder.

clip-level feature map size. For this reason, VisTR needs long training schedules (approx. 500 epochs and 1000 GPU hours) to achieve optimal performance.

The main issue of applying transformer multi-head attention is that it would consider all possible spatial and temporal locations in attention computation. Inspired by Deformable DETR [6], which applies deformable attention to attend to small sample of key sampling points around a reference point, we extend the idea to spatio-temporal deformable attention that is necessary to tackle video instance segmentation. The spatio-temporal deformable attention module only attends to a small number of fixed key sampling points around a query (reference) point, covering both the spatial and temporal directions. By decreasing the number of key elements for each query element from HWT to a small fixed number of keys K, we can mitigate the issues of slow convergence.

3.2. Spatio-Temporal Deformable Attention

A crucial component to achieving video instance segmentation is to effectively model both spatial context and temporal context. Incorporating it, a query element is a 3-d reference point that can sample a fixed set of key points in a spatio-temporal set of $H * W * T$ points.

Let q index a query element at a 3-d reference point p_q, where p_q indexes the horizontal, vertical and temporal positional information. Let z_q represent the context feature of q. Then, the spatio-temporal deformable attention (STDeformAttn) feature is computed as follows

$$STDeformAttn(\boldsymbol{z}_q, \boldsymbol{p}_q, \boldsymbol{x}) = \sum_{m=1}^{M} \boldsymbol{W}_m [\sum_{k=1}^{K} A_{mqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}(\boldsymbol{p}_q + \Delta \boldsymbol{p}_{mqk})]. \quad (2)$$

where k indexes the sampled keys and K is the total sampled key number ($K << HWT$) and m indexes the attention head. The scalar weight A_{mqk} denote the attention weight of the k^{th} sampling point in the m^{th} attention head, where A_{mqk} is normalized by $\Sigma_{k=1}^{K} A_{mqk} = 1$. A_{mqk} are predicted via linear projection over the query feature $\boldsymbol{z}_q$. $\Delta \boldsymbol{p}_{mqk}$ denote the sampling offset of the k^{th} sampling point in the m^{th} attention head. $\Delta \boldsymbol{p}_{mqk} \in \mathbb{R}^3$ are of 3-d real numbers with unconstrained range. As $\boldsymbol{p}_q + \Delta \boldsymbol{p}_{mqk}$ is fractional, bilinear interpolation is applied as in computing $\boldsymbol{x}(\boldsymbol{p}_q + \Delta \boldsymbol{p}_{mqk})$. $\Delta \boldsymbol{p}_{mqk}$ is obtained via linear projection over the query feature $\boldsymbol{z}_q$.

The computational complexity of the STDeformAttn module is of $O(2N_qC^2 + N_qMKC)$, where N_q is the number of query elements. Replacing the attention module in VisTR encoder with STDeformAttn module, where $N_q = HWT$ and $MK > C$, the complexity becomes $O(HWTCK)$. So, the computational complexity drops from $O(H^2W^2T^2C)$ to linear complexity in the spatial-temporal feature size. When it is applied to VisTR decoder the complexity becomes $O(NKC)$, where $N_q = N$ (N is the number of object queries). The proposed STDeformAttn module becomes equivalent to VisTR Transformer attention module if $K = HWT$.

3.3. Deformable Transformer Encoder

We replace the Transformer attention modules in the VisTR encoder and decoder with the proposed STDeformAttn module (see Fig. 1). Both the input and output of the encoder are same resolution spatio-temporal feature maps. In encoder, we extract features using ResNet [12] backbone. These features are transformed to contain C channels by a 1×1 convolution. In application of the STDeformAttn module in encoder, the output are spatio-temporal features with same resolutions as the input. Both the query and key elements are of pixels from the spatio-temporal feature maps. For each query pixel, the reference point is itself. To identify which spatio-temporal position the feature corresponds to, precise position information is needed. To incorporate this, we use fixed positional encoding information that contains the three dimensional (horizontal, vertical and temporal) positional information in the clip, to supplement the features.

3.4. Deformable Transformer Decoder

Transformer decoder consists of cross-attention and self-attention modules. In cross-attention modules, the object queries are the query elements, while feature maps from the encoder are used as key elements. In the self-attention modules, object queries are both query and key elements. Since our proposed STDeformAttn module is primarily designed for processing convolutional feature maps, we only replace each cross-attention module to be the STDeformAttn module.

By replacing the Transformer attention modules with STDeformAttn module in transformer of VisTR, we establish an efficient and fast converging video instance segmentation system, termed as Deformable VisTR (see Fig. 1).

4. EXPERIMENTS AND RESULTS

4.1. Datasets

YouTubeVIS dataset [2] is first dataset for video instance segmentation which contains 2238 training, 302 validation, and 343 test video clips. It contains 40 object categories along with per frame instance masks. The evaluation metrics are average precision (AP) with the video intersection over Union (IoU) of the mask sequences as the threshold.

4.2. Implementation Details

We use ImageNet [13] pre-trained ResNet-50 [12] without FPN [14] as the backbone for ablations. $M = 8$, $K = 32$, $C = 384$ are set for STDeformAttn module by default and is initialized randomly. Unless mentioned otherwise, we mainly follow VisTR [1] in training strategy, loss functions, and hyper parameter setting. The models are trained for 50 epochs and we decay learning rate by 0.1 at the 40th epoch. Following VisTR, we use Adam optimizer with backbones's learning rate as 1×10^{-5}, base learning rate of 1×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and the weight decay of 10^{-4}. We multiply the learning rates for the linear modules used for predicting object query sampling offsets by 0.1. Model is trained on 4 NVIDIA Tesla V100 GPUs of 16GB each and run time is evaluated on a single GPU.

4.3. Comparison with VisTR

As shown in Table 1, Deformable VisTR achieves on par performance as VisTR with 10× less training GPU hours and epochs. This shows that our proposed STDeformAttn module can mitigate the issue of convergence successfully.

4.4. Comparison with State of the Art

Table 2 provides comparison with state-of-the-art methods. We do not compare with [3, 15] due to their heavy component design, which results in low inference (< 6 FPS) speed. In Table 2, VisTR and Deformable VisTR are end-to-end trainable frameworks, while other methods are not. Our method achieves 34.6% mAP. Overall, our method is a fully end-to-end framework that converges 10× times faster than VisTR.

Method	Fully End-to-End	Aug.	FPS	AP
MaskTrack [2] $_{CVPR'19}$			28.6	30.3
SipMask [] $_{ECCV'20}$		✓	34.1	33.7
STEm-Seg [4] $_{ECCV'20}$		✓✓	4.4	30.6
CompFeat [16] $_{AAAI'21}$		✓✓	32.8	35.3
SG-Net [17] $_{CVPR'21}$		✓✓	19.8	34.8
STMask [18] $_{CVPR'21}$			28.6	33.5
CrossVIS [19] $_{ICCV'21}$			**39.8**	34.8
QueryInst [20] $_{ICCV'21}$			32.3	34.6
VisTR [1] $_{CVPR'21}$	✓	✓	30.0	**35.6**
Deformable VisTR	✓	✓	33.0	34.6

Table 2. Comparison of Deformable VisTR with the state-of-the-art methods on Youtube VIS val set. All the entries use ResNet-50 [12] as backbone. The methods are listed in temporal order. "✓" indicates multi-scale input images during training. "✓✓" indicates stronger data augmentation (*e.g.,* additional data [16, 4], random crop[3])

4.5. Ablation Study on STDeformAttn module

Table 3, presents ablations of varying the number of sampling points K for each reference point in the proposed STDeformAttn module. Increasing the number of sampling points K from 16 to 32 can further improve accuracy by 0.8 mAP.

backbone	K	AP
ResNet-50	16	33.8
ResNet-50	32	34.6

Table 3. Ablation of STDeformAttn module. K is the number of key points for each query feature. $K = 32$ gives the best result.

5. CONCLUSION

Deformable VisTR is designed to be an efficient, fast-converging and end-end trainable video instance segmentation framework. To implement Deformable VisTR, we propose spatio-temporal deformable attention modules, which is an efficient attention mechanism in processing clip-level feature maps. We achieve comparable performance as VisTR with 10× less GPU training hours.

Acknowledgement. This work is supported in part by a gift grant from OPPO and National Science Foundation Grant CNS1951952.

6. REFERENCES

[1] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia, "End-to-end video instance segmentation with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8741–8750.

[2] Linjie Yang, Yuchen Fan, and Ning Xu, "Video instance segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5188–5197.

[3] Gedas Bertasius and Lorenzo Torresani, "Classifying, segmenting, and tracking object instances in video with mask propagation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9739–9748.

[4] Ali Athar, Sabarinath Mahadevan, Aljosa Osep, Laura Leal-Taixé, and Bastian Leibe, "Stem-seg: Spatio-temporal embeddings for instance segmentation in videos," in *European Conference on Computer Vision*. Springer, 2020, pp. 158–177.

[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei, "Deformable convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 764–773.

[6] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision*. Springer, 2020, pp. 213–229.

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[9] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al., "Hybrid task cascade for instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4974–4983.

[10] Xizhou Zhu, Dazhi Cheng, Zheng Zhang, Stephen Lin, and Jifeng Dai, "An empirical study of spatial attention mechanisms in deep networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6688–6697.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[14] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[15] Huaijia Lin, Ruizheng Wu, Shu Liu, Jiangbo Lu, and Jiaya Jia, "Video instance segmentation with a propose-reduce paradigm," *arXiv preprint arXiv:2103.13746*, 2021.

[16] Yang Fu, Linjie Yang, Ding Liu, Thomas S Huang, and Humphrey Shi, "Compfeat: Comprehensive feature aggregation for video instance segmentation," *arXiv preprint arXiv:2012.03400*, 2020.

[17] Dongfang Liu, Yiming Cui, Wenbo Tan, and Yingjie Chen, "Sg-net: Spatial granularity network for one-stage video instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9816–9825.

[18] Minghan Li, Shuai Li, Lida Li, and Lei Zhang, "Spatial feature calibration and temporal fusion for effective one-stage video instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11215–11224.

[19] Shusheng Yang, Yuxin Fang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu, "Crossover learning for fast online video instance segmentation," *arXiv preprint arXiv:2104.05970*, 2021.

[20] Xinggang Wang Yu Li Chen Fang Ying Shan Bin Feng Yuxin Fang, Shusheng Yang and Wenyu Liu, "Instances as queries," in *Proceedings of the IEEE international conference on computer vision*, 2021.