SSTVOS: Sparse Spatiotemporal Transformers for Video Object Segmentation

Brendan Duke[1,4,5*]   Abdalla Ahmed[4]   Christian Wolf[3]   Parham Aarabi[1,4]   Graham W. Taylor[2,5]

[1]University of Toronto   [2]University of Guelph   [3]Université de Lyon, INSA-Lyon, LIRIS

[4]Modiface, Inc.   [5]Vector Institute

Abstract

In this paper we introduce a Transformer-based approach to video object segmentation (VOS). To address compounding error and scalability issues of prior work, we propose a scalable, end-to-end method for VOS called Sparse Spatiotemporal Transformers (SST). SST extracts per-pixel representations for each object in a video using sparse attention over spatiotemporal features. Our attention-based formulation for VOS allows a model to learn to attend over a history of multiple frames and provides suitable inductive bias for performing correspondence-like computations necessary for solving motion segmentation. We demonstrate the effectiveness of attention-based over recurrent networks in the spatiotemporal domain. Our method achieves competitive results on YouTube-VOS and DAVIS 2017 with improved scalability and robustness to occlusions compared with the state of the art. Code is available at https://github.com/dukebw/SSTVOS.

1. Introduction

Video object segmentation (VOS) involves simultaneous tracking and segmentation of one or more objects throughout a video clip. VOS is a challenging task in which algorithms must overcome object appearance changes, occlusion and disocclusion, as well as distinguish similar objects in motion over time.

A highly performant VOS system is important in downstream tracking applications where pixelwise tracking information is useful, such as player tracking in sports analytics, person tracking in security footage, and car and road obstacle tracking in self-driving vehicle applications. VOS methods are also relevant in interactive annotation of video data, where annotator time can be used more efficiently by using automatic video object segmentation in an annotate-predict-refine loop. Our work uses VOS as a proxy task to investigate scalable algorithms for extracting spatiotempo-
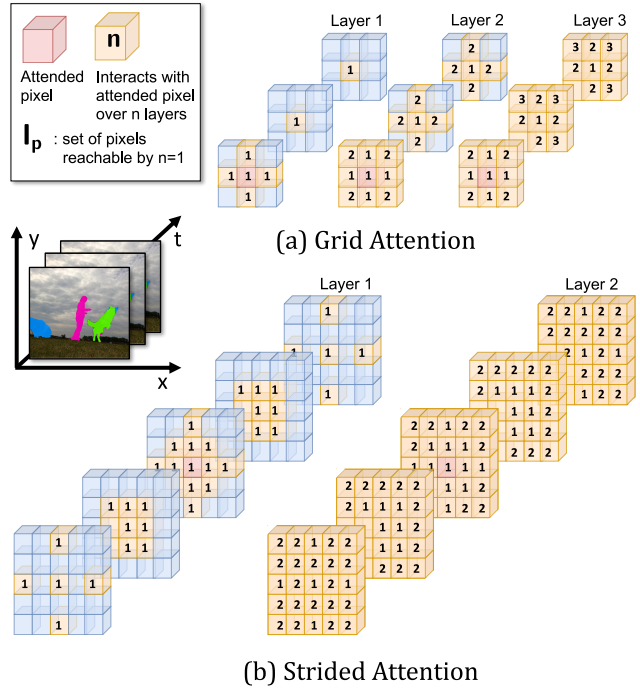

(a) Grid Attention

(b) Strided Attention

Figure 1: We propose a Transformer-based model for video object segmentation featuring self-attention over time and over space. To segment an output frame, the model learns to look up similar regions in the temporal history and to search for reference masks. We address the high computational complexity of the problem with a sparse Transformer formulation, which allows each cell to attend to each other cell over one or multiple hops. Here, interactions propagate from a given feature cell via our sparse spatiotemporal attention variants: (a) grid attention, and (b) strided attention.

ral representations from video in general, and these algorithms can be re-used for yet other video prediction tasks.

Previous methods that attempt to solve VOS can be divided into three major categories: online finetuning, mask refinement, and temporal feature propagation. Each of these categories, reviewed in detail in §2, has inherent drawbacks.

---

*Corresponding Author: brendanw.duke@gmail.com

Online finetuning methods cannot adapt to changes in object appearance throughout a sequence. The dominant mask refinement and temporal feature propagation methods are recurrent. Due to their sequential nature, recurrent methods for VOS exhibit compounding error over time, and are not parallelizable across a single example.

Motivated by the success of Transformer architectures in NLP (see §2) we propose a novel method for semi-supervised VOS that overcomes the drawbacks of online finetuning and sequential methods. Our method, Sparse Spatiotemporal Transformers (SST), processes videos in a single feedforward pass of an efficient attention-based network. At every layer of this net, each spatiotemporal feature vector simultaneously interacts with all other feature vectors in the video. SST does not require online finetuning, although it may benefit from this practice at the cost of additional runtime. Furthermore, since SST is feedforward, it avoids the compounding error issue inherent in recurrent methods. Finally, SST is fully parallelizable across a single example and can therefore take advantage of the scalability of current and future compute architectures.

Applying spatiotemporal attention operators to VOS raises two challenges: computational complexity and distinguishing foreground objects. Naïve spatiotemporal attention is square in the dimensionality of the video feature tensor, i.e., $O((THW)^2 C)$, C=# of channels. We resolve this computational complexity issue with sparse attention operators, of which we compare two promising candidates.

SST reduces feature matching FLOPs by an order of magnitude, and achieves an overall score of 81.8 on the official YouTube-VOS 2019 validation set, comparing favourably with prior work. Furthermore, we observed qualitatively improved robustness to occlusions using SST's temporal buffer of preceding frame embeddings.

Contributions — We propose a Transformer-based model for VOS, and link its inductive bias to correspondence calculations. While there is work on Transformers for representation learning in video [28, 39], these models attend over time and not densely over space. There is also recent work that adapts Transformers to video action recognition [14], however, we are unaware of work that uses Transformers in VOS, which requires dense predictions. We also contribute empirical evaluation of Transformer models applied to VOS, and argue superiority over recurrent models.

We address computational complexity using sparse attention operator variants, making it possible to apply self-attention on high-resolution videos. We extend sparse attention variants to video so that they can be used for VOS. Specifically we extend to 3D, with two spatial axes and one temporal axis, Criss-Cross Attention [18] from 2D semantic segmentation, and Sparse Attention [6] from 1D language translation. Our sparse video attention operators are not VOS specific, and could be applied to other dense video

prediction tasks. We provide our implementation [10, 11].

2. Related Work

Our work is related to previous efforts in VOS. We are motivated by work on Transformer architectures in language translation, as well as by orthogonal work on correspondence matching in computer vision.

Video Object Segmentation — In the VOS literature, **online finetuning** approaches [2, 3, 15, 23, 26, 29, 31, 47] do one-shot, or semi-supervised, VOS by finetuning a semantic segmentation network on an initial frame. A number of methods [4, 3, 31, 47] performed VOS on independent frames using finetuning alone without explicitly modeling temporal coherence. Maninis et al. [31] built on the original usage of this approach by Caelles et al. [3] by adding instance-level semantic information, while Voigtlaender and Leibe [47] added adaptive training during the sequence. **Offline** methods must use temporal information to produce future segmentations from past frames, as done using optical flow by Jang and Kim [20] and Tsai et al. [43]. Our method is in principle able to take advantage of online finetuning to improve performance, and also performs competitively using offline training alone.

Research into temporal coherence in VOS splits into two categories: approaches that refine masks, and those that propagate temporal features.

Mask refinement approaches [16, 20, 23, 33] refine a previous mask using feedforward models. Early work [23] implemented mask refinement by a recurrent connection on the concatenated frame $t-1$ output masks and frame t RGB inputs where the recurrent connection is a VGG [38] network. An extension concatenated the feature map from the first frame [33]. Yang et al. [53] used a spatial prior on the target location, with a channel-wise attention mechanism and meta-learning to adapt the network to the object given in the first frame. Bao et al. [2] propagated masks by approximate inference in an MRF, with temporal dependencies based on optical flow, and spatial dependencies using a CNN. Optical flow has also been used to add motion information via jointly training optical flow and VOS [5, 12, 15].

Temporal feature propagation approaches [16, 19, 21, 36, 42, 51] improve upon mask refinement by increasing the expressive power of the mask feature representation. At the time of writing, all such methods have used RNNs to encode and propagate spatiotemporal representations through time. Our approach falls under the temporal feature propagation category. We use sparse attention operators to propagate features temporally in a single feedforward operation.

FEELVOS [46] is a simple and fast method for solving the VOS problem. Unlike most other VOS methods, FEELVOS trains end-to-end using a pixel-wise embedding. FEELVOS also uses global and local matching to the reference and previous frames to predict masks for the video

sequence. Our work shares similarities with FEELVOS in that both methods are end-to-end trainable and conceptually simple. Our method has the added advantages of simultaneously extracting features from multiple frames using attention, and using positional encodings to learn spatiotemporal position-aware representations.

Self-attention and Correspondence Matching — End-to-end attention networks known as Transformers [44] are a dominant current approach to a multitude of text-based natural language processing (NLP) tasks [7, 8], vision and language [40] as well as speech recognition tasks [24, 30]. Recent work has explored ties between attention heads and different reasoning functions [22, 49]. More recently, Transformers have also been applied in computer vision with success [1, 13, 50, 55]. In the context of VOS, we argue that self-attention also has the potential to overcome the shortcomings of the traditionally used recurrent methods [45]. RNNs and variants are based on a Markovian assumption, where a flat vectorial hidden state is propagated over time through the sequence. Our Transformer based model takes a history of several frames and reference or predicted masks as input and allows each output region to attend to any region in the input history. This makes the propagated representation inherently structured.

3. Method

Our proposed method for VOS consists of propagating a history of τ frames over the video sequence, and allowing the model to perform spatio-temporal attention inside this history. We argue that the proposed high-level SST architecture provides inductive bias well suited for the reasoning skills required for VOS, namely computing optical flow (attending to past similar frames) and propagating reference masks over time (attending to given frames with similar appearance). We solve the challenge of computational complexity with two variants of sparse spatiotemporal attention, the "grid" and "strided" modules.

SST Architecture — The canonical text-based Transformer architecture [44] from which we drew motivation bears both similarities and differences with our VOS architecture. Like NLP Transformers, SST consists of a hierarchy of self-attention layers that form an encoder. In contrast to the encoder of an NLP Transformer, which takes as input embeddings extracted from a text sequence, SST's encoder input consists of embeddings extracted from the frames of the video to segment. As in NLP Transformers, the SST encoder's output feeds into a decoder. However, SST's decoder is unlike NLP Transformer decoders, which consist of cross-attention layers that take the output sequence embeddings as input. Instead, SST's decoder is a generic convolutional segmentation network that takes as input a concatenated set of features: current frame embeddings, attention values produced from SST's encoder's hierarchy of atten-

tion maps, and embedding output by SST's encoder. For the purpose of fair comparison with state-of-the-art work, in §4 we use CFBI's decoder module [54].

The SST architecture (Fig. 2) takes a length T sequence of $H \times W$ RGB video frames $\mathbf{S} \in \mathbb{R}^{3 \times T \times H \times W}$ as input. From $\mathbf{S}$ a CNN feature extractor f extracts

$$\mathbf{T} = f(\mathbf{S}), \tag{1}$$

a per-frame embedding $\mathbf{T} \in \mathbb{R}^{C \times T \times H' \times W'}$ at reduced spatial resolution $H' \times W'$ and with embedding channels C. In our experiments we used a ResNet-101 as f.

In order to meet hardware resource constraints, and supposing that a given frame's relation to past frames decreases with time, any given frame embedding attends to a temporal buffer of its τ preceding frame embeddings. Denote the truncated frame embedding buffer by $\mathbf{T}_\tau$. We optionally add information about the spatiotemporal position of cells in a video tensor by the positional encoding sum

$$\widetilde{\mathbf{T}} = \mathbf{T}_\tau + \mathbf{P} \tag{2}$$

where $\mathbf{P} \in \mathbb{R}^{C \times T \times H' \times W'}$ encodes absolute position. We can encode absolute position $\mathbf{P}$ using various priors, such as sinusoids or learned embeddings [44], or as a zero tensor in the case of no positional encoding.

The SST encoder processes positionally encoded per-frame embeddings $\widetilde{\mathbf{T}}$ with L self-attention layers, adding temporal context to the video representation. The SST encoder passes two outputs to the SST decoder, the first of which is spatiotemporal features $\widetilde{\mathbf{T}}_L \in \mathbb{R}^{C \times \tau \times H' \times W'}$. A composition of L layers g_l computes features $\widetilde{\mathbf{T}}_L$ as

$$\widetilde{\mathbf{T}}_L = g_L \circ g_{L-1} \circ \cdots \circ g_1(\widetilde{\mathbf{T}}). \tag{3}$$

Each encoder layer g_l consists of a sequence of multi-head attention and spatiotemporal position-wise feedforward components combined with skip connections and normalization (Fig. 2). The output $\widetilde{\mathbf{T}}_L$ of the SST encoder feeds directly into the decoder as the representation containing spatiotemporal context.

The SST decoder's other input arises as an intermediate tensor computed by the multi-head attention component of each encoder layer. Each sparse multi-head attention operation computes an attention map, which we refer to as an object affinity tensor. The role of the object affinity tensor is to propagate segmentation information from past frames (either reference or predicted) to the future using the attention distributions of the Transformer heads. This can be seen as inductive bias for the model allowing it to more easily tie attention to semantically relevant motion. Key to the procedure are tensors $I_{\mathbf{p}}^o$, which correspond to the pixels in the sparse connectivity pattern of feature cell $\mathbf{p}$, and which belong to object o. Connectivity pattern $I_{\mathbf{p}}$ determines which other feature cells are "connected to" and thus interact with
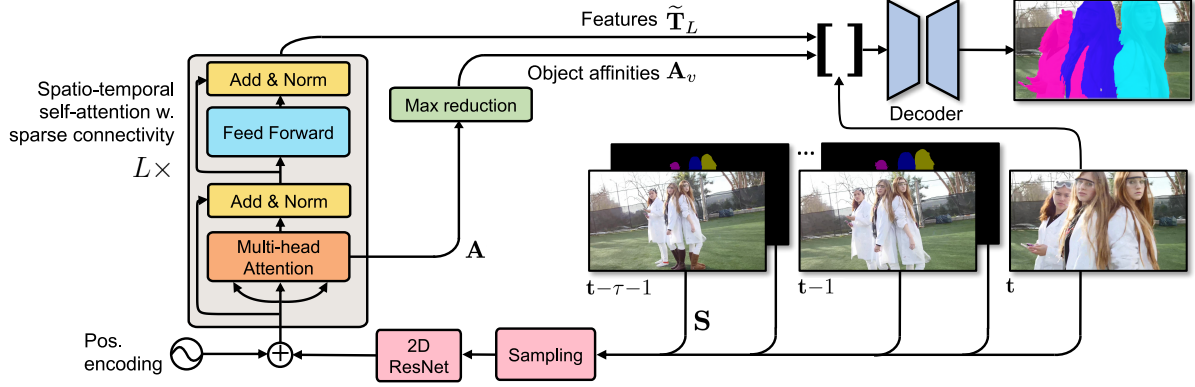
Figure 2: We propagate a history of τ frames over a video sequence and perform spatio-temporal self attention as a suitable bias for video object segmentation, allowing the model to attend to previous video frames for optical-flow like calculation, and to attend to reference frames. Computational complexity is addressed through two different sparse variants.

feature cell $\mathbf{p}$ (Fig. 1). The SST encoder uses connectivity patterns to compute the decoder's second input: object affinity values.

Object affinity values $\mathbf{A}_v \in \mathbb{R}^{L \times O \times \tau \times H' \times W'}$ represent the affinity of each of the $\tau \times H' \times W'$ cells in the spatiotemporal feature tensor with each of the O generalized objects — all reference objects plus the background. Each object affinity value $\mathbf{A}_v^{l,o}(\mathbf{p})$ is calculated as the maximum attention score in the object affinity tensor $\mathbf{A}^{l,o} \in \mathbb{R}^{|I_{\mathbf{P}}| \times \tau \times H' \times W'}$, i.e., the score of the pixel belonging to object o and most attended to by the attention head. Each of the L attention layers in the SST encoder computes its own object affinity value using a reduction operation over its object affinity tensor (Fig. 2). To enforce causality, feature cell $\mathbf{p}$'s object affinity is computed only over previous timesteps. Furthermore, each feature cell $\mathbf{p}$ attends only to feature cells in its connectivity pattern $I_{\mathbf{P}}$. We define object affinity values as

$$\mathbf{A}_v^{l,o}(\mathbf{p}) = \max_{I_{\mathbf{P}}^o \cup \{0\}} \mathbf{A}^l \qquad (4)$$

where $I_{\mathbf{P}}^o$ denotes the (possibly empty) set of feature cells belonging to object o, in connectivity pattern $I_{\mathbf{P}}$, and defaulting to zero. By taking the Tth temporal slice of object affinity values $\mathbf{A}_v$ we obtain object-discriminative features used to infer the current frame object segmentation.

Due to the form of the multi-head attention computation described next, attention maps $\mathbf{A}^l$ contain pairwise dot products between feature cells and other feature cells within their respective connectivity patterns. From these dot products we can compute Euclidean distance or normalized cross correlation. Intuitively, by doing so we use the attention map features to compare the distances or angles between per-frame embeddings in the affine subspaces projected to by each attention head. Taking all attention heads in the encoder together forms a hierarchy of such distance

(or angle) features. This improves the expressiveness of the model compared with straightforward Euclidean distance between the per-frame embeddings $\mathbf{T}$ from Equation 1.

The SST decoder (Fig. 2) is a convolutional decoder module that takes the spatiotemporal context features $\widehat{\mathbf{T}}_L$ and object affinity features $\mathbf{A}_v^{1..L}$ of all encoder layers as input. The final layer of the SST decoder produces the video object segmentation probability or masks $\mathbf{Y} \in \mathbb{R}^{H \times W}$ from the final object-discriminative representation for a given frame. It applies a scoring convolution followed by sigmoid at training time or argmax at test time. In the case of multiple objects we have probability scoremaps in $\mathbb{R}^{O \times \tau \times H \times W}$, i.e., probabilities for each generalized object (including background) for each pixel in the video. An inference protocol reduces these scoremaps to a tensor in $\mathbb{R}^{\tau \times H \times W}$ of object integer labels. We use the "naïve" inference protocol [33] and take, for each pixel, the argmax over all object probabilities including the background probability.

Sparse Attention — In this section we use T to denote a generic temporal dimension, but as described previously, we actually operate on a reduced sized buffer of length τ.

Attention is a dense operator that allows each element of a tensor to interact with all other elements at each attention layer. In VOS, attention can capture long-range dependencies without recurrence, and can be viewed intuitively as a cross-correlation operator that uses CNN features for correspondence [27], similar to prior work that used matching layers for optical flow [9].

Formally, we follow [44] in defining attention as

$$\texttt{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \texttt{softmax}(\mathbf{Q}\mathbf{K}^{\mathsf{T}})\mathbf{V}, \qquad (5)$$

where query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$ are all matrices in $\mathbb{R}^{S \times C}$ for flattened spatiotemporal dimensions $S = THW$. As we alluded to previously, we use spatiotemporal features $\mathbf{T}$ as query, key, and value, i.e., we do self-attention. Intuitively we increase the spatiotemporal context

of each feature cell $\mathbf{p}$ by doing a lookup in the spatiotemporal features connected to $\mathbf{p}$.

We adapted for VOS characteristic components of the Transformer architecture as described by Vaswani et al. [44], including multi-head attention and positional encodings. We compare the effectiveness of positional encoding schemes applied to VOS in §4. We did not normalize the softmax argument in Equation 5 by the inverse square root of channels, as we found this scaling factor reduced model effectiveness. The difference in impact of scaling factor between our VOS attention and Vaswani et al.'s NLP attention could be due to our attention operator having a comparatively low number of channels.

A computational barrier prevents naïvely using Equation 5 to perform our desired self-attention operation on spatiotemporal features $\mathbf{T}$. The attention operation given in Equation 5 is $O((THW)^2 C)$, which poses a problem for video object segmentation since for dense prediction tasks such as segmentation, model performance tends to improve with greater input resolution [56]. As an illustration of the infeasibility of using naïve attention for VOS, consider that a single layer of attention on a 16 frame video with a 64×64 feature map with 32 channels would cost more than 137 billion FLOPs, far more than the most computationally expensive CNNs in the literature at the time of writing [41].

We propose to use sparse spatiotemporal attention operators to overcome this computational barrier to applying attention for VOS. We define sparse attention operators using a connectivity pattern set $I = \{I_{\mathbf{p}_0}, \ldots, I_{\mathbf{p}_S}\}$ where $I_{\mathbf{p}}$ is a set of coordinates (i, j, k) that index a 3D tensor. Here again, connectivity pattern $I_{\mathbf{p}}$ determines which other feature cells interact with feature cell $\mathbf{p}$.

For query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$ tensors all in $\mathbb{R}^{C \times T \times H \times W}$, a sparse attention operator is defined as

$$\texttt{SparseAttn}(\mathbf{Q}, \mathbf{K}, \mathbf{V})_{\mathbf{p}} = \texttt{softmax}(\mathbf{Q}_{\mathbf{p}} \mathbf{K}_{I_{\mathbf{p}}}^{\mathsf{T}}) \mathbf{V}_{I_{\mathbf{p}}}. \quad (6)$$

We adapt two different sparse attention methods from 1D or 2D to 3D to make our attention operator computationally tractable at our desired framerate and resolution. We achieve computational tractability by careful selection of the connectivity patterns of our sparse attention operators.

Grid Attention — We adapted our first sparse attention operator from Huang et al., who also noted the computational complexity issue when applying attention for semantic segmentation [18]. In VOS, however, the computational complexity issue is exacerbated by the addition of the time dimension. We refer to the generalized operator described below as "grid attention" since the moniker "criss-cross attention" is no longer fitting in more than two dimensions.

At each layer of grid attention, each feature cell of the spatiotemporal feature tensor aggregates information from other feature cells along its X, Y, and T axes independently. Each feature cell interacts once with every other

feature cell in the spatiotemporal feature tensor that shares at least two of its X, Y, and T coordinates.

Grid attention implements Equation 6 with a connectivity pattern $I_{\mathbf{p}}$ consisting of $(T + H + W - 2)$ feature cell indices. $I_{\mathbf{p}}$ contains all feature cells along the horizontal, vertical, or temporal axes incident to location $\mathbf{p} \equiv (x, y, t)$. The grid attention weights $\texttt{softmax}(\mathbf{Q}_{\mathbf{p}} \mathbf{K}_{I_{\mathbf{p}}}^{\mathsf{T}})$ are then a matrix in $\mathbb{R}^{S \times (T+H+W-2)}$, each row of which contains weights of a convex combination. For a feature cell $\mathbf{p}$ in the spatiotemporal feature tensor, the grid attention weights are over all feature cells along $\mathbf{p}$'s temporal, vertical, and horizontal axes. By multiplying by the grid attention weights we attend over $\mathbf{p}$'s grid connectivity pattern. Note that we implemented our grid attention operator in place, so we incur no overhead from indexing tensors by $\mathbf{p}$.

In Figure 1 (top) we illustrate how grid attention propagates interactions from a single attended feature cell to all other feature cells in three sequential layers. The first grid attention layer propagates information to other feature cells in the same frame vertically and horizontally, and to feature cells at the same spatial location in all other frames. The second layer propagates interactions to the entire current frame, and vertical and horizontal axes in other frames. Finally, the third layer propagates information to all feature cells in the video feature tensor.

We can show that composing three applications of grid self-attention on spatiotemporal feature tensor $\mathbf{T}$ produces an output tensor where each spatiotemporal feature cell with coordinates (x, y, t) is composed of a weighted sum

$$\sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{k=1}^{T} (\mathbf{T}_{xyt}^{\mathsf{T}} \mathbf{T}_{iyt})(\mathbf{T}_{iyt}^{\mathsf{T}} \mathbf{T}_{ijt})(\mathbf{T}_{ijt}^{\mathsf{T}} \mathbf{T}_{ijk}) \mathbf{T}_{ijk} \quad (7)$$

over other feature cells in $\mathbf{T}$ with coordinates (i, j, k). Equation 7 shows that grid attention propagates information along "routes" through the spatiotemporal feature tensor. For a feature cell at position (x, y, t) to interact with another feature cell at an arbitrary position (i, j, k), interactions must propagate along a "route" composed of pairs of similar feature cells. Just as we might give travel directions through a city grid such as "first walk ten blocks North, then walk three blocks East", grid attention interactions must propagate a fixed number of feature cells in the X, Y, and T directions, in some order, before connecting the source feature cell with its target feature cell.

By replacing dense attention with grid attention we reduced the computational complexity of video attention from $O(C(THW)^2)$ to $O(C(T+H+W)THW)$, achieving our goal of making attention tractable for video.

Strided Attention — We investigated strided attention as an alternative sparse attention method in addition to grid attention. Drawing inspiration from sparse Transformers for sequences [6], information propagates by following paths

Model	MACs (GFLOPs)	Slowdown (%)	Params (M)
DeepLab-v3	255.4	-	66.5
Matching (CFBI)	99.6	39.0	0
SST (Local)	5.34	2.1	0.3
SST (Strided)	1.89	0.7	0.3
SST (Grid)	1.45	0.6	0.3
Naïve Attention	170.1	66.6	0.3

Table 1: Runtime and parameter analysis.

Method	O-Ft	S	$\mathcal{G}$ (%)	$\mathcal{J}_{\text{seen}}$ (%)	$\mathcal{J}_{\text{unseen}}$ (%)
YouTube-VOS 2018 Validation Split					
MSK [23]	✓	✗	53.1	59.9	45.0
OnAVOS [47]	✓	✗	55.2	60.1	46.6
OSVOS [3]	✓	✗	58.8	59.8	54.2
S2S [51]	✓	✗	64.4	71.0	55.5
PReMVOS [29]	✓	✗	66.9	71.4	56.5
BoLTVOS [48]	✓	✗	71.1	71.6	64.3
RGMP [33]	✗	✓	53.8	59.5	45.2
STM [32]	✗	✓	79.4	79.7	72.8
KMN [37]	✗	✓	81.4	**81.4**	75.3
STM$^-$ [32]	✗	✗	68.2	-	-
OSMN [53]	✗	✗	51.2	60.0	40.6
S2S [51]	✗	✗	57.6	66.7	48.2
A-GAME [21]	✗	✗	66.0	66.9	61.2
CFBI [54]	✗	✗	81.4	81.1	75.3
SST (Local)	✗	✗	**81.7**	81.2	**76.0**
YouTube-VOS 2019 Validation Split					
CFBI [54]	✗	✗	81.0	80.0	**77.9**
SST (Local)	✗	✗	**81.8**	**80.9**	76.6

Table 2: Comparison with SOTA methods on YouTube-VOS [52] 2018 and 2019. Region similarity over seen ($\mathcal{J}_{\text{seen}}$) and unseen ($\mathcal{J}_{\text{unseen}}$) categories, and overall score $\mathcal{G}$ are computed as in standard benchmarks [34]. We distinguish methods by those that use online finetuning (O-Ft) and/or synthetic data (S), and those that do not.

of locations through sparse connectivity patterns in the spatiotemporal feature tensor.

We define strided attention's connectivity pattern $I_{\mathbf{p}}$ as a generalization of strided 1D attention [6] to 3D. Our strided attention uses two different connectivity patterns $I_{\mathbf{p}}^1$ and $I_{\mathbf{p}}^2$ corresponding to separate, sequential heads of multihead attention. The first "local" connectivity pattern $I_{\mathbf{p}}^1$ routes to all feature cells in a cube of side-length h from $\mathbf{p}$, i.e., $I_{\mathbf{p}}^1 = (\mathbf{p} + (l_x, l_y, l_z) : l_x, l_y, l_z < h)$. The subsequent "strided" connectivity pattern $I_{\mathbf{p}}^2$ routes to all feature cells in the video tensor that can reach $\mathbf{p}$ by taking steps of size h along each axis, i.e., $I_{\mathbf{p}}^2 = (\mathbf{p} + (l_x, l_y, l_z) : l_x, l_y, l_z \mod h = 0)$. We choose $h \approx \sqrt{H}$ to reduce the computational complexity by a square root from $O(C(THW)^2)$ to $O(C(THW)^{3/2})$. We visualize strided attention's connectivity pattern in Figure 1 (bottom).

The relative efficiency of grid and strided attention depends on the size of T, since we assume that H and W are both large relative to T. In a setup where $H, W \in \{64, 128\}$, and $T \approx 8$, strided attention costs about 1.3 to 1.4 times as many operations as grid attention.

Runtime — Table 1 provides a runtime and parameter analysis. We computed the multiply-accumulates (MACs) of SST for a 3-frame temporal buffer, input resolution of 465×465, 128 channels, and 3 Transformer layers. We show MACs in both absolute GFLOPs and as slowdown relative to DeepLab-v3 backbone MACs. We also compare to CFBI's local/global matching. Note that SST's local temporal window ($\tau = 3$) is larger than CFBI's ($\tau = 1$). Finally, we compare to naïve attention. Both naïve attention and CFBI's global matching attend pairwise to an entire feature map, costing 39.0% and 66.6% slowdown relative to DeepLab-v3's runtime. In contrast, SST factorizes the computation by attending to all other spatiotemporal feature cells over sequential layers (Fig. 1). SST reduces slowdown by more than an order of magnitude to $\approx 2\%$.

4. Experiments and Results

We present benchmark experiment results against state-of-the-art (SOTA) methods on the DAVIS 2017 [35] and YouTube-VOS [52] datasets. We further analyze the effect of different sparse attention operators, history sizes and positional encodings through ablation studies on DAVIS 2017.

YouTube-VOS — is a large scale VOS dataset comprised of 4453 YouTube video clips spanning 94 object categories [52]. YouTube-VOS includes an official validation set of 507 videos with held out labels, which can be evaluated only through an evaluation server. YouTube-VOS has been the basis of challenges in 2018 and 2019, yielding two versions of the validation set and evaluation server. The 2019 version contains new and corrected annotations. The YouTube-VOS validation set contains 26 object categories that are unique to the validation set, used to test the generalization capability of VOS models to object classes unseen in the training set. The convention is to compute region similarity $\mathcal{J}$ and contour accuracy $\mathcal{F}$ as defined by Perazzi et al. [34]. As a single metric for comparing results, it is also customary to compute overall score $\mathcal{G}$ as the average of four values comprising region similarity and contour accuracy scores for seen and unseen classes.

In Table 2 we present our model's results on YouTube-VOS 2018 and 2019 alongside previous SOTA results. Our model (SST) performs favourably against all previous methods in overall score $\mathcal{G}$, even methods that use online fine-
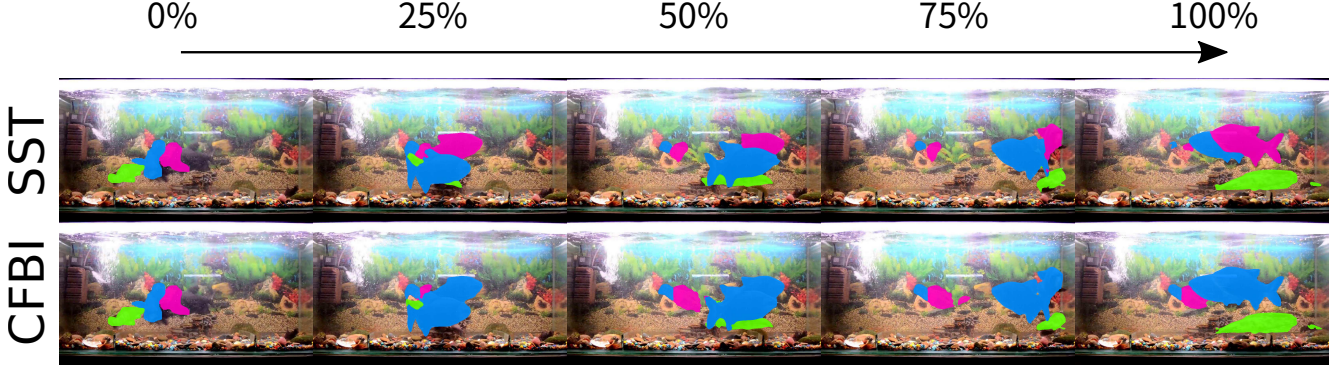
Figure 3: A qualitative example from YouTube-VOS validation showing SST handling occlusion. In this challenging example, SST's temporal history enables robust tracking of all three fish, while CFBI confuses two fish once they overlap.

Method	O-Ft	S	$\mathcal{J}\&\mathcal{F}$ (%)	$\mathcal{J}$ (%)	$\mathcal{F}$ (%)
DAVIS 2017 Validation Split					
OSVOS-S [31]	✓	✓	68.0	64.7	71.3
OSVOS [3]	✓	✗	60.3	56.6	63.9
OnAVOS [47]	✓	✗	65.4	61.6	69.1
CINM [2]	✓	✗	70.6	67.2	74.0
PReMVOS [29]	✓	✗	77.7	73.9	<u>81.7</u>
RGMP [33]	✗	✓	66.7	64.8	68.6
STM$^-$ [32]	✗	✓	71.6	69.2	74.0
STM† [32]	✗	✓	81.8	79.2	84.3
KMN [37]	✗	✓	76.0	74.2	77.8
OSMN [53]	✗	✗	54.8	52.5	57.1
FAVOS [4]	✗	✗	58.2	54.6	61.8
VM [17]	✗	✗	-	56.5	-
DyeNet [26]	✗	✗	69.1	67.3	71.0
A-GAME† [21]	✗	✗	70.0	67.2	72.7
FEELVOS† [46]	✗	✗	71.5	69.1	74.0
CFBI [54]	✗	✗	74.9	72.1	77.7
CFBI† [54]	✗	✗	81.9	79.3	84.5
SST (Local)	✗	✗	<u>78.4</u>	<u>75.4</u>	81.4
SST (Local)†	✗	✗	**82.5**	**79.9**	**85.1**

Table 3: Comparison with SOTA methods on DAVIS 2017 [35]. We denote online finetuning methods by O-Ft, and synthetic data methods by S. We report results trained only on the DAVIS 2017 training set, and pre-trained on YouTube-VOS. YouTube-VOS pre-training is denoted by †.

tuning (denoted by O-Ft) and pre-training on synthetic data (denoted by S). Note that our unique method performs competitively against recurrent methods that have undergone multiple research and development cycles where one method builds on the foundation of another, for example [21] extends [33], which in turn extends [23].

DAVIS 2017 — is the latest dataset in the DAVIS initiative to promote VOS research. DAVIS 2017 comprises 150 sequences, which include 376 separately annotated objects [35]. We additionally evaluate our method on DAVIS 2017 [35], and compare our results with SOTA in Table 3. We report our DAVIS results following the traditionally used region similarity $\mathcal{J}$ and contour accuracy $\mathcal{F}$ metrics as well as their mean $\mathcal{J}\&\mathcal{F}$. Our DAVIS 2017 evaluation provides additional experimental evidence that SST performs favourably compared with existing SOTA methods, since SST achieves a mean $\mathcal{J}\&\mathcal{F}$ score of 78.4, whereas previous SOTA scored a $\mathcal{J}\&\mathcal{F}$ of 74.9 under a comparable experimental setup (without online finetuning or synthetic data).

We evaluate only on DAVIS 2017 and not DAVIS 2016 [34] because DAVIS 2017 is strictly a more challenging superset of DAVIS 2016. Furthermore DAVIS 2016 contains only single object annotations and therefore we could make only limited evaluation of SST's ability to handle multi-object context using DAVIS 2016.

Ablation Studies — Figure 3 shows a qualitative example on the YouTube-VOS validation set of SST handling foreground occlusion, where one fish entirely occludes another before the second fish becomes disoccluded again.

We used DAVIS 2017 to perform ablation studies on interesting components of our method, including sparse attention operators, positional encodings, and temporal history size. We first describe and compare different design choices for our positional encodings.

We investigated sinusoidal positional encodings for the temporal dimension, as used in Transformers for language translation [44]. We hypothesized that sinusoidal positional encodings would be superior to learned positional embeddings because of the data imbalance of temporal positions in VOS datasets, which are skewed towards lower frame numbers. Sinusoidal positional encodings can be interpolated or extrapolated to generalize to underrepresented absolute frame numbers, whereas positional embeddings have no such generalization mechanism.

We present our positional encoding results in Table 4 us-

Positional Encoding	$\mathcal{J}\&\mathcal{F}$ (%)
None	73 ± 2
Sinusoidal	75.6 ± 0.6

Table 4: Positional encodings on DAVIS 2017 val.

τ	$\mathcal{J}\&\mathcal{F}$ (%)
1	75.8 ± 0.3
2	76.2 ± 0.3
3	76.5 ± 0.3

Table 5: Temporal histories τ on DAVIS 2017 val.

Sparse Attention	Layers	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
Grid	1	65.3	62.3	68.4
Grid	2	66.1	62.6	69.5
Grid	3	64.2	61.0	67.4
Local	2	76.2	72.8	79.5
Strided	2	69.5	65.7	73.3
LocalStrided	2	72.3	69.1	75.6

Table 6: SST sparse attention variants on DAVIS 2017 val.

ing the $\mathcal{J}\&\mathcal{F}$ score on the DAVIS 2017 validation set. The positional encoding labeled "None" is our baseline attention with no positional information, while "Sinusoidal" uses sinusoidal positional encodings for all spatiotemporal dimensions X, Y, and T. To evaluate robustness to hyperparameters, we computed the mean and variance of $\mathcal{J}\&\mathcal{F}$ over multiple runs for each positional encoding scheme, varying the number of Transformer layers and the temporal history. Sinusoidal temporal positional encodings performed best, achieving both a higher mean score and lower variance. The superiority of positional encodings supports our hypothesis that information about distance-from-reference is important in positional encodings for VOS. The lower variance indicates that sinusoidal positional encodings form a robust prior for finding correspondences in VOS.

In Table 5 we evaluate the effect of increasing SST's temporal history τ. We varied the temporal history while keeping other hyperparameters fixed, and computed the variance over multiple runs. We observed that even a modest increase in temporal history improves the $\mathcal{J}\&\mathcal{F}$ score. We expect that efficiency improvements further increasing the temporal history size will improve the effectiveness of SST's temporal history mechanism even further.

In Table 6 we compare the performance of SST using different sparse attention variants. We expected that increasing the number of layers would improve grid attention's performance due to the increasing receptive field of each feature cell. A larger receptive field should improve the effectiveness of the object affinity value from multi-head attention. Grid attention's $\mathcal{J}\&\mathcal{F}$ score increased as expected from one to two layers, but dropped off for three layers possibly due to overfitting. We also expected that local attention should be effective for DAVIS 2017's fast framerate compared to strided attention, both of which we describe in §3. LocalStrided attention provides a tradeoff between the local and global context windows of local and strided attention, respectively. So that LocalStrided attention can attend transitively to all feature cells in just two consecutive sparse attention layers, we set the kernel size equal to the square root of the feature tensor width. For fair comparison, we kept the same kernel size for all strided attention variants. In general, local and strided attention outperformed grid attention, showing that local and strided connectivity patterns form superior priors for VOS compared with grid attention.

Discussion — We present a method for VOS purely based on end-to-end attention. Future work could be analogous to Transformer models' progression on language translation tasks, where researchers successfully applied Transformers to increasingly long sequences. For example, Dai et al. combined recurrence with attention to translate arbitrary-length sequences [7], and Kitaev et al. introduced locality-sensitive hashing instead of dot-product attention, to reduce computational complexity from squared to $O(N \log N)$ while using reversible networks to model arbitrary-length sequences with constant memory usage [25]. In order to evaluate VOS on long sequences the VOS community would have to overcome a dataset creation challenge, since the current benchmark dataset YouTube-VOS contains sequences with at most 36 labeled frames, sampled at 6 fps. We propose that future work could use interactive and semi-automatic annotation methods, based on the existing high-quality VOS models, to create datasets with longer and therefore more challenging sequences.

5. Conclusions

We presented Sparse Spatiotemporal Transformers (SST), which, up to our knowledge, constitutes the first application of an entirely attention-based model for video object segmentation (VOS). We evaluated the positive effect of positional encodings and the advantage of attending over a history of multiple frames, suggesting a superiority of a spatiotemporally structured representation over the flat hidden representation of recurrent models. We showed that SST is capable of state-of-the-art results on the benchmark VOS dataset YouTube-VOS, attaining an overall score of $\mathcal{G} = 81.8$, while having superior runtime scalability compared with previous state of the art, including methods based on recurrence. We provide code [11] to reproduce all experiments in our work, including sparse video-attention operator implementations [10], so that the community can build on the promising idea of using attention-based models for video. Open challenges are the memory requirements inherent in a model with only weak Markovian assumptions, which for the moment prevents the increase of history size to a desirable longer extent.

References

[1] Anonymous. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *openreview ICLR 2020 submissino*, 2020.

[2] Linchao Bao, Baoyuan Wu, and Wei Liu. CNN in MRF: video object segmentation via inference in A cnn-based higher-order spatio-temporal MRF. In *CVPR*, 2018.

[3] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, 2017.

[4] Jingchun Cheng, Yi-Hsuan Tsai, Wei-Chih Hung, Shengjin Wang, and Ming-Hsuan Yang. Fast and accurate online video object segmentation via tracking parts. In *CVPR*, 2018.

[5] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. Segflow: Joint learning for video object segmentation and optical flow. In *ICCV*, 2017.

[6] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. In *arXiv:1904.10509*, 2019.

[7] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

[9] 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 *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[10] Brendan Duke. Sparse spatiotemporal transformer. https://github.com/dukebw/sparse-spatiotemporal-transformer, 2021.

[11] Brendan Duke. SSTVOS. https://github.com/dukebw/SSTVOS, 2021.

[12] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *CVPR*, 2017.

[13] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman. Video Action Transformer Network. In *CVPR*, 2020.

[14] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *CVPR*, 2019.

[15] Ping Hu, Gang Wang, Xiangfei Kong, Jason Kuen, and Yap-Peng Tan. Motion-guided cascaded refinement network for video object segmentation. In *CVPR*, 2018.

[16] Yuan-Ting Hu, Jia-Bin Huang, and Alexander Schwing. Maskrnn: Instance level video object segmentation. In *Advances in Neural Information Processing Systems 30*, 2017.

[17] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G. Schwing. Videomatch: Matching based video object segmentation. In *ECCV*, 2018.

[18] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. 2019.

[19] Varun Jampani, Raghudeep Gadde, and Peter V. Gehler. Video propagation networks. In *CVPR*, 2017.

[20] Won-Dong Jang and Chang-Su Kim. Online video object segmentation via convolutional trident network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[21] Joakim Johnander, Martin Danelljan, Emil Brissman, Fahad Shahbaz Khan, and Michael Felsberg. A generative appearance model for end-to-end video object segmentation. In *CVPR*, 2019.

[22] C. Kervadec, T. Jaunet, G. Antipov, M. Baccouche, R. Vuillemot, and C. Wolf. How Transferrable are Reasoning Patterns in VQA? In *CVPR*, 2021.

[23] Anna Khoreva, Federico Perazzi, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017.

[24] Chanwoo Kim, Minkyu Shin, Abhinav Garg, and Dhananjaya Gowda. Improved Vocal Tract Length Perturbation for a State-of-the-Art End-to-End Speech Recognition System. In *Proc. Interspeech 2019*, 2019.

[25] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020.

[26] Xiaoxiao Li and Chen Change Loy. Video object segmentation with joint re-identification and attention-aware mask propagation. In *The European Conference on Computer Vision (ECCV)*, 2018.

[27] Jonathan L Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? In *Advances in Neural Information Processing Systems 27*, 2014.

[28] J. Lu, D. Batra, D. Parikh, and S. Lee. ViLBERT: Pre-training Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *NeurIPS*, 2019.

[29] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. In *Asian Conference on Computer Vision*, 2018.

[30] Christoph Lüscher, Eugen Beck, Kazuki Irie, Markus Kitza, Wilfried Michel, Albert Zeyer, Ralf Schlüter, and Hermann Ney. RWTH ASR Systems for LibriSpeech: Hybrid vs Attention. In *Proc. Interspeech 2019*, 2019.

[31] Kevis-Kokitsi Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. Video object segmentation without temporal information. 2018.

[32] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. *ICCV*, 2019.

[33] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *CVPR*, 2018.

[34] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016.

[35] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017.

[36] Amaia Salvador, Miriam Bellver, Manel Baradad, Ferran Marques, Jordi Torres, and Xavier Giro-i Nieto. Recurrent neural networks for semantic instance segmentation. In *arXiv:1712.00617*, 2017.

[37] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Kernelized memory network for video object segmentation. In *ECCV*, 2020.

[38] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for Large-Scale image recognition. In *International Conference on Learning Representations*, 2015.

[39] C. Sun, F. Baradel, K. Murphy, and C. Schmid. Learning Video Representations using Contrastive Bidirectional Transformer. In *arxiv:1906.05743*, 2019.

[40] Hao Tan and Mohit Bansal. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. In *EMNLP-IJCNLP*, 2019.

[41] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[42] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning video object segmentation with visual memory. In *ICCV*, 2017.

[43] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J. Black. Video segmentation via object flow. In *CVPR*, 2016.

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*. 2017.

[45] Carles Ventura, Miriam Bellver, Andreu Girbau, Amaia Salvador, Ferran Marques, and Xavier Giro-i Nieto. Rvos: End-to-end recurrent network for video object segmentation. In *CVPR*, 2019.

[46] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *CVPR*, 2019.

[47] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *BMVC*, 2017.

[48] Paul Voigtlaender, Jonathon Luiten, and Bastian Leibe. BoLTVOS: Box-Level Tracking for Video Object Segmentation. *arXiv:1904.04552*, 2019.

[49] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[50] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local Neural Networks. In *CVPR*, 2018.

[51] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *ECCV*, 2018.

[52] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas S. Huang. Youtube-vos: A large-scale video object segmentation benchmark. In *ECCV*, 2018.

[53] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K. Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, 2018.

[54] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. In *Proceedings of the European Conference on Computer Vision*, 2020.

[55] H. Zhao, J. Jia, and V. Koltun. Exploring Self-attention for Image Recognition . In *CVPR*, 2020.

[56] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *ECCV*, 2018.