





TDViT: Temporal Dilated Video Transformer for Dense Video Tasks

Guanxiong Sun^{1,2} , Yang Hua¹ , Guosheng Hu² , and Neil Robertson¹ 

¹ EEECS/ECIT, Queen’s University Belfast, UK

² Oosto, Belfast, UK

{gsun02, y.hua, n.robertson}@qub.ac.uk, huguosheng100@gmail.com

Abstract. Deep video models, for example, 3D CNNs or video transformers, have achieved promising performance on sparse video tasks, i.e., predicting one result per video. However, challenges arise when adapting existing deep video models to dense video tasks, i.e., predicting one result per frame. Specifically, these models are expensive for deployment, less effective when handling redundant frames and difficult to capture long-range temporal correlations. To overcome these issues, we propose a Temporal Dilated Video Transformer (TDViT) that consists of carefully-designed temporal dilated transformer blocks (TDTB). TDTB can efficiently extract spatiotemporal representations and effectively alleviate the negative effect of temporal redundancy. Furthermore, by using hierarchical TDTBs, our approach obtains an exponentially expanded temporal receptive field and therefore can model long-range dynamics. Extensive experiments are conducted on two different dense video benchmarks, i.e., ImageNet VID for video object detection and YouTube VIS for video instance segmentation. Excellent experimental results demonstrate the superior efficiency, effectiveness, and compatibility of our method. The code is available at <https://github.com/guanxionsun/TDViT>.

1 Introduction

In the past decade, 2D convolutional neural network (CNN) based architectures [22,38,39,40,41,52] have dominated various computer vision tasks for still image understanding, e.g., image classification, object detection and semantic segmentation. Given the excellent performance on still images, 2D CNNs are adapted to video understanding by extending the 2D convolutional layer through the temporal axis, i.e., 3D CNNs [9,34,43]. Recently, the computer vision community has seen a model shift from conventional CNNs to vision transformers [27,46]. This trend began with the pioneering approach, ViT [27], which leverages Transformers [46] to understand an image as a sequence of non-overlapping patches. Some follow-up approaches focus on improving ViT in different aspects, such as finer feature representations [20,32,47], better token generations [51,57], more efficient training processes [42], etc. Not surprisingly, the great success of transformers on still-image tasks also leads to investigation of the use of transformers for video tasks [2,5,33].

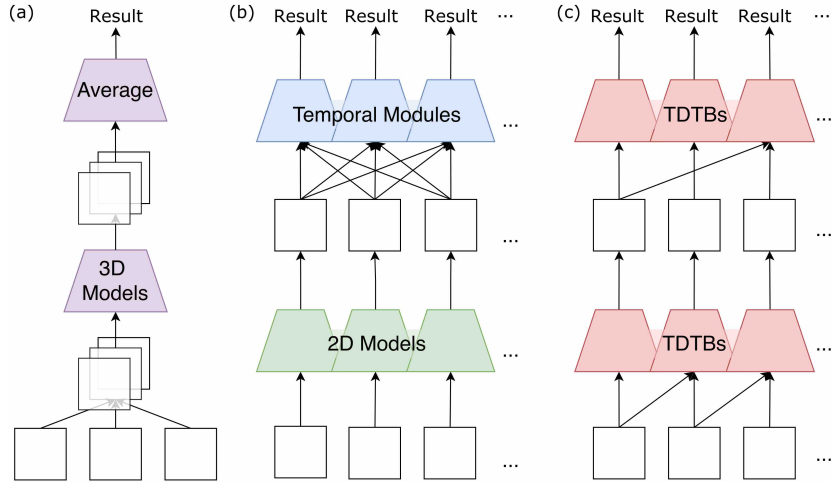


Fig. 1. Architecture (a) is widely used in sparse video tasks. 3D models, e.g., 3D CNNs, take multiple frames as input and generate one output by averaging the spatiotemporal representations. Architecture (b) is used for dense video tasks. Considering the computational cost, 2D models are used to extract features of independent frames, and then temporal modules, e.g., correlation filters, are leveraged to model spatiotemporal correspondences. Our TDViT (c) is designed for dense video tasks, which can efficiently and effectively extract spatiotemporal representations using temporal dilated transformer blocks (TDTB). *Best viewed in colour.*

Most video models (3D CNNs [9,17,43] or video transformers [5,14,33]) are designed for *sparse* video tasks, e.g., video classification [1,25,44] and action recognition [9,16,18], whose goal is to predict one label for one given video. These models utilise 3D operations, e.g., 3D convolutional layers or spatiotemporal self-attentions, to learn a spatiotemporal representation for a video. The architecture of sparse video models is shown in Fig. 1(a). Differently, *dense* video tasks aim to predict one result per frame, e.g., video object detection [10,37] and video instance segmentation [6,54]. However, problems arise when integrating video models into dense video tasks. Specifically, these models are (1) computationally intensive, resulting in high deployment costs; (2) unable to extract relevant information from redundant frames; and (3) challenging to capture long-range temporal correlations. In practise, therefore, state-of-the-art approaches for dense video tasks are still built in a hybrid manner: extracting spatial features using 2D models and adding temporal modules [13,46] to generate spatiotemporal representations. The architecture of sparse models is shown in Fig. 1(b).

In this paper, we propose a Temporal Dilated Video Transformer (TDViT) whose overall architecture is shown in Fig. 1(c). Inspired by visual transformers

[27,32,42] which are naturally suitable for sequence modelling, we exploit transformers to extract spatiotemporal features. Unlike most video models that are based on self-attention transformers, we present a novel temporal dilated transformer block (TDTB) with two distinct designs. Firstly, a memory structure that stores features of previous frames is introduced to each TDTB. During inference, the query tokens are derived from the current frame for each time step, whereas the key and value tokens are sampled from the memory structure. In this way, TDTB can extract multi-frame spatiotemporal features with a single-frame overhead. Secondly, a temporal dilation factor is proposed to control the temporal frequency of feature sampling by skipping neighbouring frames. Therefore, TDTB can alleviate the negative effects of video redundancy. Furthermore, by employing hierarchical TDTBs in stages, TDViT acquires an exponentially expanded temporal receptive field. Here, the temporal receptive field represents the number of video frames considered when extracting spatiotemporal features. As a result, TDViT can effectively capture long-range temporal correlations and achieve higher accuracy.

To better demonstrate the differences between our TDViT and other visual models, we categorise existing visual models based on their architectures and their target tasks. For example, ResNets [22,52] and Inceptions [39,40] are CNN-based models designed for image tasks. VSwin [33] and TSFMR [5] are transformer-based models designed for sparse video tasks. Our approach is also a transformer based model but designed for dense video tasks. The details and categorisation of existing video models are presented in Table 1.

To sum up, our contributions are listed as follows:

- Unlike the existing hybrid solutions (2D image models equipped with temporal modules [13,46]) for dense video tasks, we propose a neat and compact architecture, Temporal Dilated Video Transformers (TDViT), for spatiotemporal representation learning.
- We design a novel temporal dilated transformer block (TDTB) that can efficiently extract spatiotemporal representations and alleviate the negative effect of video redundancy by cooperating with a memory structure and temporal dilated feature sampling process. Furthermore, by employing hierarchical TDTBs, our approach can effectively capture long-range temporal correlations.
- We conduct extensive experiments on two datasets for different dense video tasks, i.e., ImageNet VID for video object detection and YouTube VIS for video instance segmentation. Excellent experimental results demonstrate the superior effectiveness, efficiency, and compatibility of our method. Given these advantages, we believe that our TDViT can serve as a general-purpose backbone for various dense video tasks.

2 Related Work

Video Models for Sparse Video Tasks. For video modelling, 3D CNNs are investigated in pioneering approaches [9,34,43]. These methods extend 2D

Table 1. Categorization of visual models based on their architectures and their target tasks. For dense video tasks, most SOTA methods use 2D models as their backbone and leverage temporal modules, e.g., attention [46] or correlation filter [13] to get spatiotemporal representations. To the best of our knowledge, our approach is the first end-to-end backbone designed for dense video tasks.

Arch.	2D Models	3D Sparse Models	3D Dense Models
CNN	ResNet [22,52]	R3D [45]	2D models + Temporal modules [13,46]
	Inception [39,40]	I3D [9]	
	EffNet [41]	SFNet [17]	
Transformer	ViT [27]	Non-local [48]	TDViT (Ours)
	DeiT [42]	TimeSformer [5]	
	Swin [32]	VSwin [33]	

CNNs through the temporal axis, but the performance is limited by their small temporal receptive fields. To address this, multi-head attentions [46] are introduced to fuse spatiotemporal representations. For example, NLNet [48] adopts self-attentions to model pixel-level dependency globally. GCNet [7] presents a lightweight global context block and achieves good performance with less computation. DNL [56] captures better visual clues using disentangled NL blocks.

Dense Video Tasks. Dense video tasks aim to output one prediction per frame. For video object detection, DFF [60], FGFA [59] and THP [58] use optical flow to propagate or aggregate feature maps of neighbour frames. Recent methods, SELSA [50], RDN [12] and MEGA [10], perform proposal-level aggregation by modelling relationships between objects using attention mechanisms [46]. For video instance segmentation, MaskTrack R-CNN [54] extends instance segmentation from image domain to video domain. MaskProp [4] uses mask propagation to obtain smoother results. StemSeg [3] treats the video clip as 3D spatial-temporal volume and segments instances by clustering methods. TransVOD [23] and VisTR [49] adapt DETR [8] for VOD and VIS tasks, respectively, in a query-based end-to-end fashion. These methods are built in a hybrid manner, extracting features with 2D models and relying on additional temporal modules. In contrast, our TDViT is a neat and compact model that is built with unified TDTBs.

Vision Transformers. Our approach is inspired by the recent trend of attention based vision transformers. Among these methods, ViT [27] is the first to use consecutive transformers to replace the CNN backbones. It reshapes an image into a sequence of non-overlapping flattened 2D patches and achieves a good speed-accuracy trade-off for image classification. Many recent papers improve the performance of ViT in different aspects. For example, DeiT [42] addresses the issue that ViT requires a large amount of data by knowledge distillation. TiT [20] and PVT [47] generate multi-scale feature maps to capture finer visual

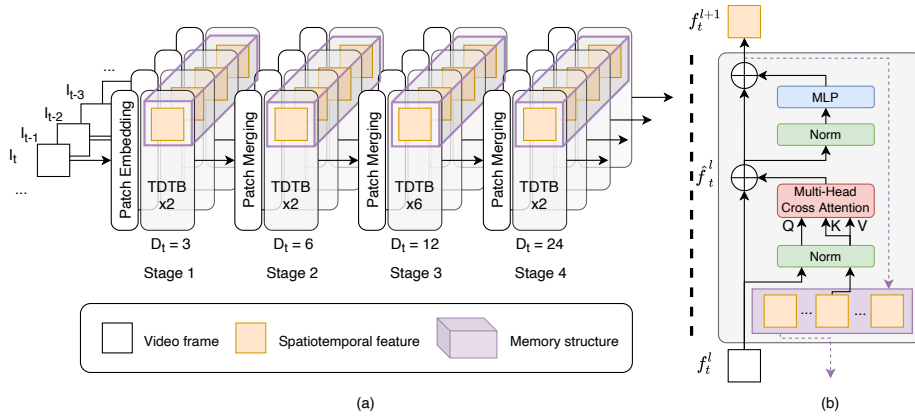


Fig. 2. (a) Overview. TDViT contains four stages which consist of several temporal dilated transformer blocks (TDTB). A memory structure (purple cuboids) is introduced into the TDTB, which stores features of previous frames (yellow rectangles) and enables our approach to dynamically establish temporal connections. The temporal dilation factor D_t is used to control the memory sampling process and reduce the video redundancy. (b) Details of a TDTB. For every time step, the query tokens are from the current frame I_t but the key and value tokens are derived from memory sampling. Finally, the memory structure saves the output features and deletes the oldest features. *Best viewed in colour.*

clues. T2T [57] uses a small transformer network to generate better token representations. The most related work to ours is the Swin [32], which uses local attention to gradually reduce the resolution of feature maps and achieves impressive performance in various vision tasks. These methods demonstrate that transformers do well in modelling spatial dependencies. Inspired by these methods, we dive deeper into the essence of transformers and find that they are naturally suitable for modelling the correlations of sequence data.

3 Method

3.1 Overall Architecture

An overview of our TDViT is presented in Fig. 2(a). Our approach consists of four stages. Each stage is obtained by stacking several temporal dilated transformer blocks (TDTB). Each TDTB has a memory structure to store and sample features of previous frames. TDViT takes a frame I_t of one video $\{I_\tau\}_{\tau=1}^T$ as input, where T is the length of the video and t denotes the current time step. The previous frames of the video can be denoted as $\{I_\tau\}_{\tau=1}^{t-1}$. By modelling the spatiotemporal correlations between I_t and $\{I_\tau\}_{\tau=1}^{t-1}$, TDViT can output more accurate predictions for every frame.

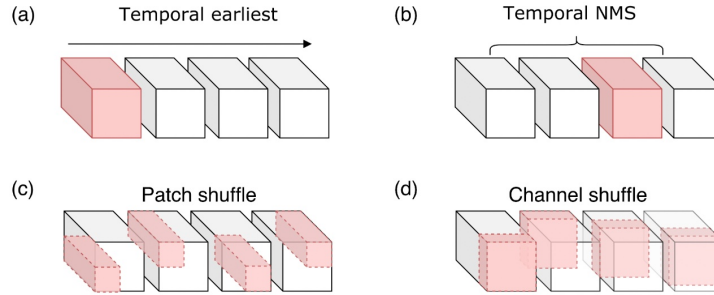


Fig. 3. Illustration of memory sampling. The red cuboid denotes the sampled features and the grey cuboid denotes stored features in memory. *Best viewed in colour.*

3.2 Temporal Dilated Transformer Block

Inspired by the dilated convolutions which introduce spatial skip steps for feature aggregation, we propose the temporal dilated transformer block to construct skip connections in the temporal dimension. Specifically, unlike existing self-attention based transformer modules, the temporal dilated transformer block (TDTB) has two novel designs: (1) a memory structure \mathcal{M} to explicitly store previous features; (2) a temporal dilation factor D_t to control the stride of temporal connections. The input of a TDTB is the features of the current frame, which can be in different forms, such as feature maps with arbitrary sizes or proposal features. Given $D_t=d$, the l -th TDTB samples features from $\{I\tau\}_{t-d}^t$ as reference features f^R . Then I_t is enhanced with f^R using multi-head cross attention (MCA). Specifically, the tokens for query Q , key K and value V are obtained by normalising features of the current frame f_t^l and reference features f^R , respectively. A 2-layer MLP with GELU non-linearity is applied to the MCA’s output features \hat{f}_t^l . Two LayerNorm LN layers are applied before MCA and MLP, respectively. The output feature f_t^{l+1} is obtained by applying residual connections after MCA and MLP layers. Formally,

$$\begin{aligned}
 f^R &= \text{Sampling}(\mathcal{M}, D_t), \\
 \hat{f}_t^l &= \text{MCA}(\text{LN}(f_t^l), \text{LN}(f^R)) + f_t^l, \\
 f_t^{l+1} &= \text{MLP}(\text{LN}(\hat{f}_t^l)) + \hat{f}_t^l,
 \end{aligned} \tag{1}$$

where $\text{Sampling}(\cdot, \cdot)$ denotes the sampling process guided by D_t . An illustration of TDTB is shown in Fig. 2 (b).

Memory Sampling and Feature Reuse. The memory structure \mathcal{M} is updated at every time step by replacing the oldest frame with f_t^{l+1} . Given a temporal dilation D_t , we design different strategies to sample reference features f^R . The temporal *earliest* strategy selects the oldest feature map in the memory. The temporal NMS selects the feature map that has the maximum L2 norm.

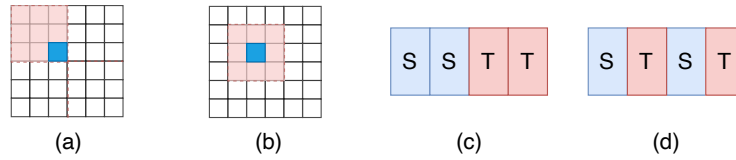


Fig. 4. (a) Window and (b) correlation based local attentions. The blue rectangle denotes one query token, the red dotted rectangle denotes the range of participated key tokens. (c) Split and (d) factorised spatiotemporal schemes. The blue and red boxes denote space-only self-attention and TDTB, respectively. *Best viewed in colour.*

The *patch*-wise shuffle strategy first splits the feature maps into four windows, i.e., top left, top right, bottom left, and bottom right. Then, it randomly selects four time steps from four groups, respectively, and combines them as the reference feature map. Similarly, the *channel*-wise shuffle strategy split the feature maps into C groups according to channels, where C is the number of channels. Among these strategies, the first two are feature-level operations, while the last two are more fine-grained group-wise operations. The differences between these strategies are shown in Fig. 3.

Once we obtain the sample features f^R , we reuse the feature for D_t time steps and then sample new f^R from the updated memory. Instead of computing query, key, and value tokens for every frame, TDTB only extracts query tokens from the current frame and reuses f^R to obtain key and value tokens. As a result, TDViT achieves a slightly faster speed compared to its 2D counterpart, the Swin [32].

Efficient Local Attentions. The most well-known transformers are built with multi-head self-attention (MSA) modules in which one query token attends to all key tokens globally. We exploit local attentions because they have lower computational costs. In addition, local attentions are very suitable for dense video tasks since the movement of an object is continuous and should be inside a local region for a short time. Specifically, we introduce two different local attentions. Firstly, following the protocol of [32], we adapt the window-based local attention (Fig. 4(a)), where key tokens are from reference frames within the same window of the query token. Secondly, inspired by [13,19,24], we design a local attention based on correlation (Fig. 4(b)), where key tokens are from reference frames within a neighbour region of the query token.

3.3 Spatiotemporal Attention Schemes

We investigate different spatiotemporal attention schemes, i.e., the order of the space block and the temporal block (TDTB) in a stage. Inspired by [5,33], we design two spatiotemporal schemes: split and factorised schemes. The split scheme adds several temporal blocks (TDTB) after consecutive space blocks (Fig. 4(c)),

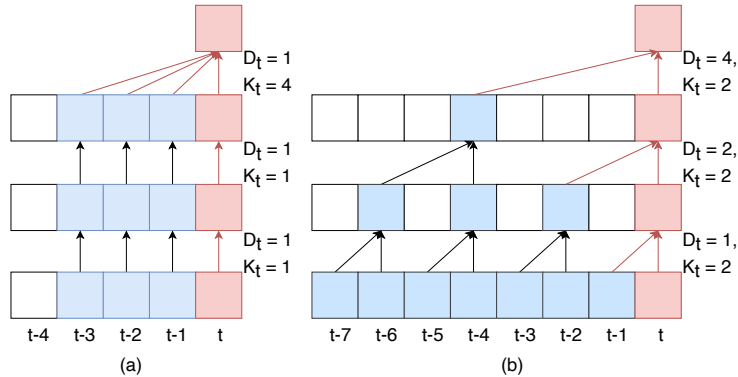


Fig. 5. Illustration of how hierarchical TDTBs increase the temporal receptive field. (a) and (b) show the framework of RDN [12] and our TDViT, respectively. The red and blue rectangles denote the current frame and frames within the temporal receptive field, respectively. *Best viewed in colour.*

while the factorised scheme adds one TDTB after each space block as shown in Fig. 4(d).

3.4 Temporal Receptive Field of Hierarchical TDTB

One TDTB with temporal dilation D_t can aggregate temporal information from time step $t - D_t$ to t . We use TDTBs in multiple hierarchical stages to obtain an exponentially expanded temporal receptive field. For a better demonstration, we introduce the temporal kernel size K_t which denotes the number of frames considered in an aggregation process. For example, an aggregation process with $K_t=2$ and $D_t=4$ aggregates 2 frames whose temporal distance is 4. Using K_t and D_t , we can describe the process of TDViT and other SOTA methods for VOD from a unified perspective. Fig. 5(a) shows a simplified version of RDN [12] where $D_t=\{1, 1, 1\}$ and $K_t=\{1, 1, 4\}$ from bottom to top. For RDN, temporal aggregation only happens in the detection head after the backbone feature extraction. TDViT hierarchically fuses temporal features onto the current frame. Fig. 5(b) shows a simplified version of TDViT, where $D_t=\{1, 2, 4\}$ and $K_t=\{2, 2, 2\}$. With similar complexity, i.e., the times of feature aggregation shown as the number of red connections in Fig. 5, the temporal receptive fields of RDN and TDViT are 4 and 8, respectively.

3.5 Architecture and Variants

Following the design principles of Swin [32], we build three basic variants of our models, called TDViT-T (tiny), TDViT-S (small) and TDViT-B (base), whose sizes and computational complexities are similar to Swin-T, Swin-S and Swin-B. The query dimension of each transformer head is 32, and the expansion layer of

each MLP is 4. Two hyperparameters are introduced to describe each variant: C denotes dimensions of the linear embedding in the first stage and L denotes the number of transformers in each stage. The *basic* variants are represented as TDViT-T: $C=96, L=(2,2,6,2)$; TDViT-S: $C=96, L=(2,2,18,2)$; TDViT-B: $C=128, L=(2,2,18,2)$. By adding two additional TDTBs at the end of stage 3, we further obtain the *advanced* variants which can be represented as TDViT-T⁺: $C=96, L=(2,2,8,2)$; TDViT-S⁺: $C=96, L=(2,2,20,2)$; TDViT-B⁺: $C=128, L=(2,2,20,2)$.

4 Experiments

To verify the generality of our TDViT, we conduct experiments on two datasets, i.e., ImageNet VID [36] for video object detection (VOD) and YouTube VIS [54] for video instance segmentation (VIS). Firstly, we compare TDViT with other backbones. Then, we adapt TDViT to SOTA VOD and VIS methods to evaluate its compatibility. Finally, we conduct ablation studies on different designs, as mentioned in the Section 3. We report detailed evaluation results in the COCO [30] format. Specifically, we use the average precision (AP^{box}) metric which computes the average precision over ten IoU thresholds [0.5: 0.05: 0.95] for all categories. Meanwhile, we report other important metrics, e.g., AP_{50}^{box} and AP_{75}^{box} are calculated at IoU thresholds of 0.50 and 0.75 [15], and AP_S^{box} , AP_M^{box} , AP_L^{box} are calculated on different object scales, i.e., small, medium and large. All training and testing are performed on Tesla V100 GPUs.

Video Object Detection Setup. The ImageNetVID dataset [36] contains 3,862 training and 555 validation videos. Following previous approaches [37,50,59], we add overlapped 30 classes of the ImageNet DET dataset into the training set. Specifically, we sample 15 frames from each video in the VID training set and at most 2,000 images per class from the DET training set. We employ an AdamW [26] optimizer for 3 epochs in total, with an initial learning rate of 10^{-3} and 10^{-4} after the 2nd epoch, using a weight decay of 0.05. We follow the augmentation and regularisation strategies of Swin [32]. Given the key frame I_k , we randomly sample four frames from $\{I_\tau\}_{k-D_t}^{k+D_t}$ to approximately form the memory of TDTBs in four stages, respectively. During training, detection losses are only computed for the key frame I_k .

Video Instance Segmentation Setup. We also evaluate TDViT for video instance segmentation on YouTube VIS 2019 dataset [54] and report the validation results as [6,3]. There are 3,471 training videos and 507 validation videos. For training and testing, the original images are resized to 640×360 . The metrics are computed on masks, denoted as AP^{mask} and AP_{75}^{mask} , etc.

4.1 Video Object Detection

We first compare TDViT with other backbones using single-frame FasterRCNN [35] as the base detector. Then, we integrate TDViT with the SELSA [50] video

Table 2. Comparisons with other widely used backbones for video object detection.

Backbone	Arch.	AP ^{box}	AP ₅₀ ^{box}	AP ₇₅ ^{box}	AP _S ^{box}	AP _M ^{box}	AP _L ^{box}	#Param	FPS
R-50 [22]	CNN	44.3	72.5	47.2	6.3	19.7	49.3	44.4M	23.0
R-101 [22]		48.5	75.5	53.1	7.6	23.1	53.7	63.4M	19.5
Swin-T [32]	ViT	47.1	77.2	51.5	9.5	23.2	52.4	47.8M	22.8
Swin-S [32]		52.6	82.4	59.3	10.0	26.5	58.6	69.1M	16.7
Swin-B [32]		53.2	82.7	60.4	9.1	27.7	59.0	107.1M	12.8
TDViT-T	ViT	49.1	78.5	52.7	8.0	25.6	53.1	47.8M	23.9
TDViT-T ⁺		50.9	79.9	55.7	9.1	26.9	57.2	51.3M	21.9
TDViT-S		55.4	84.1	63.4	10.3	29.3	61.2	69.1M	16.8
TDViT-S ⁺		55.7	84.1	63.2	10.5	30.4	61.4	72.7M	16.2
TDViT-B		56.0	84.4	64.1	10.7	28.0	62.3	107.1M	12.9
TDViT-B ⁺		56.1	84.7	64.2	10.0	29.9	61.8	113.5M	12.2

object detection (VOD) method and compare it with other SOTA VOD methods.

Comparisons with other backbones. We compare TDViT with various widely used backbones, e.g., R-50 and R-101 as baselines. We also introduce a transformer-based backbone, Swin [32] as the baseline for fair comparisons. Detailed evaluation results are tested with the single-frame FasterRCNN [35] as shown in Table 2. All TDViT variants greatly surpass other architectures with comparable model sizes, i.e., number of parameters denoted as #Param. For *basic* variants, TDViT-T achieves 49.1% of AP^{box}, outperforming R-50/Swin-T (44.3/47.1%) by +4.8/2.0% of AP^{box}, respectively. TDViT-S/B achieves 55.4/56.0% of AP^{box}, outperforming Swin-S/B (52.6/53.2%) by +2.8% and TDViT-S outperforms R-101 (48.5%) by +6.9% of AP^{box}. Using the *advanced* variants, i.e., TDViT-T⁺/S⁺/B⁺, the performance is improved to 50.9/55.7/56.1% of AP₅₀^{box}, respectively. To compare the run-time speed, we test all models on the same device, and the detailed results (FPS) are listed in the last column of Table 2. TDViT achieves better speed-accuracy trade-offs, and the speed-accuracy curve is shown in Fig. S1 in the supplementary material.

Comparisons with SOTA. SOTA VOD methods use multi-frame proposal aggregation to learn the relationships of objects in a video and achieve promising performance. We use SELSA [50] as our baseline for simplicity. In our re-implementation of SELSA, we introduce the distillation strategy proposed in RDN [12] where the top 75 out of total 300 proposals on the reference frames are selected for conducting aggregation. We follow previous methods [10,12,37,59] and use Average Precision with IoU threshold 0.5 (AP₅₀^{box}) as the default evaluation metric. Following the protocol in [59], we use motion IoU (*mIoU*) to split the dataset into three groups: slow (*mIoU* < 0.7), medium (0.7 < *mIoU* < 0.9) and fast (*mIoU* > 0.9), and report AP₅₀^{box} on three subsets, respectively. As

Table 3. Comparisons with SOTA video object detection methods.

Methods	Backbone	AP ₅₀ ^{box}	Slow	Medium	Fast	FPS	Hardware
FGFA [59]	R-101	76.3	83.5	75.8	57.6	1.2	K40
PSLA [19]	R-101+DCN	80.0	-	-	-	26.0	Titan V
CHP [53]	R-101	76.7	-	-	-	37.0	-
TransVOD [23]	R-101	81.9	-	-	-	-	-
SELSA [50]	R-101	80.3	86.9	78.9	61.4	-	-
RDN [12]	R-101	81.8	89.0	80.0	59.5	10.6	Tesla V100
MEGA [10]	R-101	82.9	89.4	81.6	62.7	8.7	2080 Ti
TFBlender [11]	R-101	83.8	-	-	-	4.9	2080 Ti
MAMBA [37]	R-101	84.6	-	-	-	9.1	Titan RTX
SELSA*	R-101	81.5	87.8	79.9	65.4	14.5	Tesla V100
SELSA+Ours	TDViT-T	83.9	88.6	83.8	67.7	16.2	Tesla V100
SELSA+Ours	TDViT-T ⁺	84.5	88.9	84.3	69.4	15.7	Tesla V100

Table 4. Comparisons with other backbones for video instance segmentation.

Backbone	Arch.	AP ^{mask}	AP ₅₀ ^{mask}	AP ₇₅ ^{mask}	AP _S ^{mask}	AP _M ^{mask}	AP _L ^{mask}	#Param	FPS
R-50 [22]	CNN	30.3	50.0	32.1	11.8	30.5	40.6	58.1M	18.0
R-101 [22]		32.9	52.3	35.6	9.8	33.0	44.1	77.1M	17.2
Swin-T [32]	ViT	33.7	57.1	35.1	12.8	33.5	44.2	61.5M	18.6
Swin-S [32]		36.8	60.9	40.1	16.0	37.3	48.8	82.8M	14.3
TDViT-T	ViT	35.4	59.6	38.6	13.6	36.9	46.4	61.5M	18.6
TDViT-T ⁺		36.1	60.2	39.9	14.0	37.5	48.1	65.0M	18.0
TDViT-S		37.7	61.9	40.5	16.9	37.9	49.6	82.8M	14.3
TDViT-S ⁺		38.2	62.5	41.6	17.6	38.1	51.2	86.3M	13.9

shown in Table 3, our reimplemented SELSA with R-101 [22] achieves 81.5% of AP₅₀^{box} at 14.5 FPS. Then, we replace the R-101 backbone with our TDViT-T, and the performance is improved by 2.4% to 83.9% of AP₅₀^{box} at 16.2 FPS. We further introduce the advanced variant TDViT⁺, and our method achieves 84.5% of AP₅₀^{box} at 15.7 FPS. Compared to other SOTA methods, our method achieves a better speed-accuracy trade-off. For example, MAMBA [37] and TFBlender+MEGA [11] achieve 84.6% and 83.8% of AP₅₀^{box}, respectively, but they run at relatively slow speeds, 9.1 and 4.9 FPS.

4.2 Video Instance Segmentation

Similar to the experiment designs in video object detection, we first compare TDViT with other backbones using MaskTrack R-CNN [54] as the base segmentor. Then, we compare the performance with other SOTA video instance segmentation methods.

Table 5. Comparisons of our approach with SOTA methods for video instance segmentation.

Methods	Backbone	AP^{mask}	AP_{50}^{mask}	AP_{75}^{mask}	FPS	Hardware
Stem-Seg [3]	R-50	30.6	50.7	33.5	12.1	Titan RTX
SipMask [6]	R-50	33.7	54.1	35.8	28.0	Titan RTX
SG-Net [31]	R-50	34.8	56.1	36.8	22.9	Titan RTX
TFBlender [11]	R-50	35.7	57.1	37.6	21.3	Titan RTX
CrossVis [55]	R-50	36.3	56.8	38.9	-	-
VisTR [49]	R-50	36.2	59.8	36.9	30.0	Tesla V100
MaskTrack R-CNN* [54]	R-50	30.3	50.0	32.1	18.0	Tesla V100
MaskTrack R-CNN+Ours	TDViT-T	35.4	59.6	38.6	18.6	Tesla V100
MaskTrack R-CNN+Ours	TDViT-T ⁺	36.1	60.2	39.9	18.0	Tesla V100

Comparisons with other backbones. We compare our TDViT-T with other backbones with similar model sizes, i.e., number of parameters (#Param) in Table 4. The pre-trained Mask R-CNN [21] model on the COCO [30] dataset is used to initialise the network. We first report results using R-50 and R-101 backbones. Then, we replace the backbone with a stronger Swin model [32]. Compared with using R-50/101, the performance of using Swin-T/S is improved by 3.4/3.9% to 33.7/36.8% of AP^{mask} , respectively. Then, we replace the backbone with TDViT. Compared to Swin-T/S, the performance of TDViT-T/S is improved by 1.7/0.9% to 35.4/37.7% of AP^{mask} , respectively. By introducing the advanced variants, TDViT-T⁺/S⁺, the performance is further improved to 36.1/38.2% of AP^{mask} . We also test the run-time speed of these backbones. As shown in Table 4, our method can achieve better speed-accuracy trade-offs.

Comparisons with SOTA. We compare our method with SOTA video instance segmentation methods and report detailed results in Table 5. We use MaskTrack R-CNN [54] as the base segmentor and use * to denote our reimplemented version. Using R-50 as the backbone, MaskTrack R-CNN * achieves 30.3% AP^{mask} at 18.0, which is the baseline. By replacing the backbone with TDViT-T/T⁺, the performance is improved to 35.4/36.1% of AP^{mask} and runs at 18.6/18.0 FPS, respectively. The performance and speed are comparable to recent SOTA methods, e.g., VisTR [49] and CrossVis [55].

4.3 Ablation Study

Spatiotemporal attention schemes. We conduct experiments on the split and factorised schemes. Detailed orders of space-only blocks and temporal blocks (TDTB) inside the stage 3 are listed in Table 6. Other stages contain 2 blocks, and we use one space-only block followed by one TDTB. More details of spatiotemporal schemes on different variants are shown in the supplementary material. The split scheme has 0.5% higher AP_{50}^{box} than the factorised scheme, and

Table 6. Effect of different spatiotemporal attention schemes.

Scheme	Backbone	Stage 3		AP ₅₀ ^{box}
		Details	Depth	
Space-only	Swin-T	[s]*6	6	77.2
Factorised scheme	TDViT-T	[s, t]*3	6	78.0
Split scheme	TDViT-T	s*3, t*3	6	78.5
Split scheme	TDViT-T ⁺	s*3, t*5	8	79.9

Table 7. Effect of different temporal dilations. The four numbers denote the temporal dilations in four stages, respectively.

$\{D_t\}$	AP ₅₀ ^{box}	AP ₇₅ ^{box}	AP _S ^{box}	AP _M ^{box}	AP _L ^{box}
(1,2,4,8)	77.2	51.5	8.6	23.2	52.4
(2,4,8,16)	77.9	52.0	8.6	23.4	53.0
(3,6,12,24)	78.1	52.1	8.7	23.8	52.2
(4,8,16,32)	78.5	52.7	8.0	25.6	53.1
(5,10,20,40)	78.4	52.5	8.5	24.9	53.4

thus we use the split scheme by default. With a increased number of TDTBs, e.g., from 6 to 8, TDViT-T⁺ achieves better performance with 79.9% of AP₅₀^{box}.

Temporal dilations. Larger temporal dilations enable TDViT to fuse temporal information over a longer time span. However, larger temporal dilations increase the difficulty of learning the spatiotemporal correspondences. In this part, we conduct experiments to find a good trade-off between the large temporal receptive field and the learning difficulty. We gradually increase the temporal dilations from the lower stage to the higher stage. The specific number of temporal dilations and the corresponding results are listed in Table 7. Temporal dilations of four stages are denoted as $\{D_t\}$. Performance improves when we increase the temporal dilations. In our experiments, with temporal dilations $\{D_t\} = \{4, 8, 16, 32\}$, our method achieves the optimal performance 78.5% of AP₅₀^{box} and we use this setting by default. The performance improvement is saturated when we further increase the temporal dilations. For example, with $\{D_t\} = \{5, 10, 20, 40\}$, the performance drops by 0.1% of AP₅₀^{box}.

Memory sampling strategies. In the TDTB, a good sampling strategy should be able to generate informative features from the memory structure. For example, given the temporal dilation $D_t = 16$, the sampled features should well represent 16 neighbour frames. We design different sampling strategies and conduct experiments to find a better solution. Their differences are illustrated in §3.2 and Fig. 3. The detailed results are shown in Table 8. Among these strategies, patch shuffle achieves the best performance 78.8% of AP₅₀^{box} because the sampled

Table 8. Comparisons between different sampling strategies

	AP_{50}^{box}	AP_{75}^{box}	AP_S^{box}	AP_M^{box}	AP_L^{box}	FPS
Temporal earliest	78.5	52.7	8.0	25.6	53.1	23.9
Temporal NMS	78.5	52.6	8.3	25.7	53.0	19.9
Channel shuffle	77.7	51.4	8.4	25.1	52.1	20.2
Patch shuffle	78.8	53.0	8.5	25.7	53.3	19.7

Table 9. Comparisons of local attentions according to region sizes.

Size	Window	Correlation	AP_{50}^{box}	FPS
7×7		✓	78.8	18.4
7×7	✓		78.5	23.9
5×5	✓		77.6	25.2
9×9	✓		78.1	22.1

features have good diversities. However, considering the run-time speed, we use the temporal earliest strategy by default, which achieves a comparable accuracy with 78.5% of AP_{50}^{box} but a much faster speed at 23.9 FPS.

Efficient local attention. Local attention is important for TDViT to achieve a good speed-accuracy trade-off. We conduct experiments on different local attentions. Firstly, the window based local attention, which is used in Swin [32]. The feature maps are split into non-overlapping windows and then attentions are computed within each window locally. Secondly, the correlation based local attention, which is used in PSLA [19], optical flow [13] and tracking methods [28,29]. Given one query token, local attentions are computed within a small region around the query location. The differences between these two designs are illustrated in §3.2 and Fig. 4. We also test with different local region sizes. The detailed settings and results are presented in Table 9. By default, we use the window based attention with size 7×7 for a good speed-accuracy trade-off.

5 Conclusion

We present a Temporal Dilated Video Transformer (TDViT) for dense video tasks. The key component in TDViT is the temporal dilated transformer block (TDTB), which can obtain accurate multi-frame spatiotemporal representations with a single-frame computational cost and effectively sample useful information from redundant frames. Moreover, by using hierarchical TDTBs, our approach can capture long-range temporal correlations, further improving accuracy. TDViT achieves excellent speed-accuracy trade-offs on the ImageNet VID and YouTube VIS datasets and is compatible with different frameworks. The simplicity and strong performance suggest that TDViT can potentially serve as a general-purpose backbone for various dense video tasks.

References

1. Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., Vijayanarasimhan, S.: Youtube-8m: A large-scale video classification benchmark. arXiv preprint arXiv:1609.08675 (2016)
2. Arnab, A., Deghani, M., Heigold, G., Sun, C., Lučić, M., Schmid, C.: Vivit: A video vision transformer. In: ICCV (2021)
3. Athar, A., Mahadevan, S., Osep, A., Leal-Taixé, L., Leibe, B.: Stem-seg: Spatio-temporal embeddings for instance segmentation in videos. In: ECCV (2020)
4. Bertasius, G., Torresani, L.: Classifying, segmenting, and tracking object instances in video with mask propagation. In: CVPR (2020)
5. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding? In: ICML (2021)
6. Cao, J., Anwer, R.M., Cholakkal, H., Khan, F.S., Pang, Y., Shao, L.: Sipmask: Spatial information preservation for fast image and video instance segmentation. In: ECCV (2020)
7. Cao, Y., Xu, J., Lin, S.C.F., Wei, F., Hu, H.: Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In: ICCVW (2019)
8. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV (2020)
9. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR (2017)
10. Chen, Y., Cao, Y., Hu, H., Wang, L.: Memory enhanced global-local aggregation for video object detection. In: CVPR (2020)
11. Cui, Y., Yan, L., Cao, Z., Liu, D.: Tf-blender: Temporal feature blender for video object detection. In: ICCV (2021)
12. Deng, J., Pan, Y., Yao, T., Zhou, W., Li, H., Mei, T.: Relation distillation networks for video object detection. In: ICCV (2019)
13. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: ICCV (2015)
14. Duan, H., Zhao, Y., Xiong, Y., Liu, W., Lin, D.: Omni-sourced webly-supervised learning for video recognition. In: ECCV (2020)
15. Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J.M., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV **88**, 303–338 (2009)
16. Fabian Caba Heilbron, Victor Escorcia, B.G., Niebles, J.C.: Activitynet: A large-scale video benchmark for human activity understanding. In: CVPR (2015)
17. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: ICCV (2019)
18. Goyal, R., Kahou, S.E., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Freund, I., Yianilos, P., Mueller-Freitag, M., Hoppe, F., Thureau, C., Bax, I., Memisevic, R.: The “something something” video database for learning and evaluating visual common sense. In: ICCV (2017)
19. Guo, C., Fan, B., Gu, J., Zhang, Q., Xiang, S., Prinet, V., Pan, C.: Progressive sparse local attention for video object detection. In: ICCV (2019)
20. Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., Wang, Y.: Transformer in transformer. In: NeurIPS (2021)
21. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV (2017)
22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)

23. He, L., Zhou, Q., Li, X., Niu, L., Cheng, G., Li, X., Liu, W., Tong, Y., Ma, L., Zhang, L.: End-to-end video object detection with spatial-temporal transformers. In: ACMMM (2021)
24. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: CVPR (2017)
25. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: CVPR (2014)
26. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
27. Kolesnikov, A., Dosovitskiy, A., Weissenborn, D., Heigold, G., Uszkoreit, J., Beyer, L., Minderer, M., Dehghani, M., Houlsby, N., Gelly, S., Unterthiner, T., Zhai, X.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
28. Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J.: Siamrpn++: Evolution of siamese visual tracking with very deep networks. In: CVPR (2019)
29. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X.: High performance visual tracking with siamese region proposal network. In: CVPR (2018)
30. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
31. Liu, D., Cui, Y., Tan, W., Chen, Y.: Sg-net: Spatial granularity network for one-stage video instance segmentation. In: CVPR (2021)
32. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV (2021)
33. Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., Hu, H.: Video swin transformer. arXiv preprint arXiv:2106.13230 (2021)
34. Qiu, Z., Yao, T., Mei, T.: Learning spatio-temporal representation with pseudo-3d residual networks. In: ICCV (2017)
35. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NeurIPS (2015)
36. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. IJCV **115**(3), 211–252 (2015)
37. Sun, G., Hua, Y., Hu, G., Robertson, N.: Mamba: Multi-level aggregation via memory bank for video object detection. In: AAAI (2021)
38. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI (2017)
39. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR (2015)
40. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR (2016)
41. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: ICML (2019)
42. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: ICML (2021)
43. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: ICCV (2015)
44. Tran, D., Wang, H., Torresani, L., Feiszli, M.: Video classification with channel-separated convolutional networks. In: ICCV (2019)
45. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: CVPR (2018)

46. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017)
47. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. arXiv preprint arXiv:2102.12122 (2021)
48. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: CVPR (2018)
49. Wang, Y., Xu, Z., Wang, X., Shen, C., Cheng, B., Shen, H., Xia, H.: End-to-end video instance segmentation with transformers. In: CVPR (2021)
50. Wu, H., Chen, Y., Wang, N., Zhang, Z.: Sequence level semantics aggregation for video object detection. In: ICCV (2019)
51. Wu, H., Xiao, B., Codella, N.C.F., Liu, M., Dai, X., Yuan, L., Zhang, L.: Cvt: Introducing convolutions to vision transformers. In: ICCV (2021)
52. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: CVPR (2017)
53. Xu, Z., Hrustic, E., Vivet, D.: Centernet heatmap propagation for real-time video object detection. In: ECCV (2020)
54. Yang, L., Fan, Y., Xu, N.: Video instance segmentation. In: ICCV (2019)
55. Yang, S., Fang, Y., Wang, X., Li, Y., Fang, C., Shan, Y., Feng, B., Liu, W.: Crossover learning for fast online video instance segmentation. In: ICCV (2021)
56. Yin, M., Yao, Z., Cao, Y., Li, X., Zhang, Z., Lin, S., Hu, H.: Disentangled non-local neural networks. In: ECCV (2020)
57. Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z., Tay, F.E.H., Feng, J., Yan, S.: Tokens-to-token vit: Training vision transformers from scratch on imagenet. In: ICCV (2021)
58. Zhu, X., Dai, J., Yuan, L., Wei, Y.: Towards high performance video object detection. In: CVPR (2018)
59. Zhu, X., Wang, Y., Dai, J., Yuan, L., Wei, Y.: Flow-guided feature aggregation for video object detection. In: ICCV (2017)
60. Zhu, X., Xiong, Y., Dai, J., Yuan, L., Wei, Y.: Deep feature flow for video recognition. In: CVPR (2017)