computergraphics tübingen

Video Transformers for Classification and Captioning

Nam Nguyen The, Vojtěch Sýkora, Leon Trochelmann, Swadesh Jana, Eric Nazarenus

October 8, 2024

ML for Graphics, Vision and Language
Practical Course SS 2024
Wilhelm Schickard Institute of Computer Science

Video Transformers for Classification and Captioning

Nam Nguyen The, Vojtěch Sýkora, Leon Trochelmann, Swadesh Jana, Eric Nazarenus

Abstract

In this project, we investigated the application of transformer-based models for video classification and captioning tasks, focusing on the Self-supervised Video Transformer (SVT) and Video Mamba (VM) models. We explored the effectiveness of these models on a complex dataset featuring intricate human interactions and daily life activities. Our approach involves adapting the pre-trained SVT encoder for downstream tasks while keeping it frozen. This allows us to assess its generalization ability on other datasets while having lower computational needs. We also developed a specialized data processing pipeline to handle the unique challenges presented by the dataset. By comparing the performance of the Self-supervised Video Transformer with Video Mamba, we provide insights into the capabilities of self-supervised models in video understanding tasks. Our research contributes to the field by demonstrating the potential of transformer-based models in handling complex video data, which could reduce reliance on large labeled datasets for effective video analysis in the future. The code has been uploaded to GitHub here.

1 Introduction

The rapid evolution of transformer-based [Vas17] vision models has opened new avenues for tackling various video-related tasks. These models, mainly due to their ability to capture temporal dynamics and spatial features, have shown great promise in tasks like video classification and captioning [ADH+21, LNC+22, YXA+22]. Recent research has explored these areas; nevertheless, more experiments must be performed to understand the potential challenges better.

In the context of video classification, transformer models have been employed with increasing success. For example, recent works [ADH+21, LNC+22, YXA+22] have shown that transformers can outperform traditional convolutional neural networks (CNNs) by better modeling long-range dependencies in video sequences. Similarly, in video captioning, transformers have demonstrated the ability to generate more contextually accurate descriptions [LLL+22, WYH+22]. However, these models often struggle when applied to complex, real-world datasets that include a wide range of activities. One example is the Charades dataset [SVW+16], which presents intricate human interactions and ambiguous daily life activities.

Our research draws inspiration from two state-of-the-art video-encoder models: the Self-supervised Video Transformer (SVT) [RNK+22], and Video Mamba (VM) [LLW+24]. With its ability to learn powerful representations without relying on large labeled datasets, the SVT model is beneficial where labeled data or computational resources for downstream tasks are scarce. Furthermore, Video Mamba, known for its advanced temporal modeling capabilities, has significantly improved the understanding and processing of long video sequences.

This paper investigates the advantages of the self-supervised pre-trained SVT encoder when applied to downstream tasks. By keeping the encoder frozen, we explore its generalization ability in previously unseen datasets, allowing us to assess the effectiveness of the self-supervised model. This approach also highlights the benefits of reducing the reliance on large labeled datasets and minimizing computational costs. VM serves as a benchmark for comparing the performance of the self-supervised encoder under similar conditions. Moreover, to handle the complexities of the Charades dataset, we developed a specialized data processing pipeline designed to enhance the models' ability to interpret intricate video data.

In summary, the key contributions of this research are:

1. Adaptation of the Self-supervised Video Transformer (SVT) and Video Mamba models to video classification and captioning and extending them to tackle the complex Charades

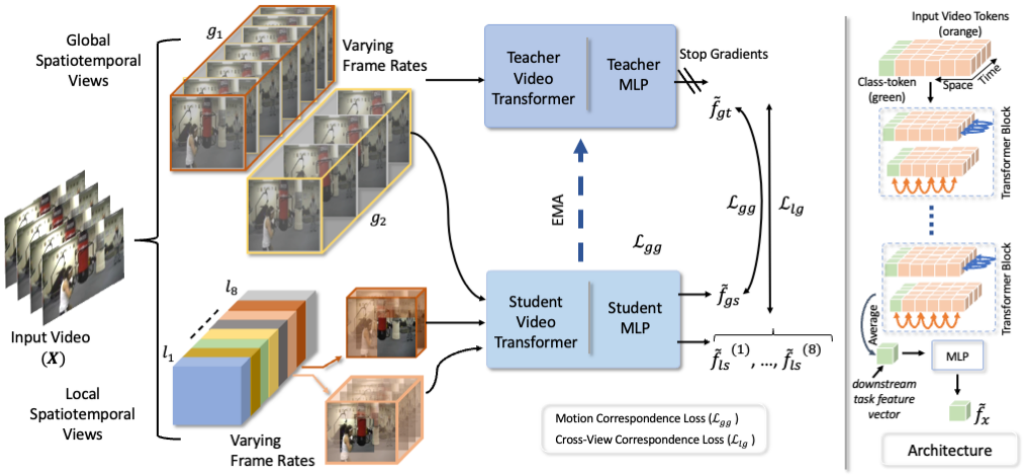


Figure 1: Self-supervised Video Transformer (SVT) architecture from [RNK+22]

dataset.

2. Development of a specialized data processing pipeline tailored to the unique characteristics of video data in the Charades dataset, along with comprehensive experimentation and evaluation to assess the performance and adaptability of the models.

The following sections first cover related works, followed by a detailed explanation of SVT and Video Mamba models. Next, we discuss the experimentation techniques and training setup, leading to a comprehensive presentation of results and discussion. Finally, the paper concludes with a summary of the key findings.

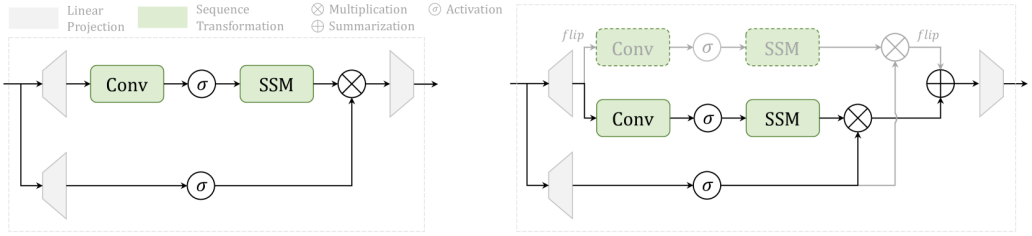
2 Related Work

This section discusses the video classification and captioning tasks performed in the experiments. The SVT and Video Mamba models used for downstream tasks are explained.

2.1 Video classification & captioning

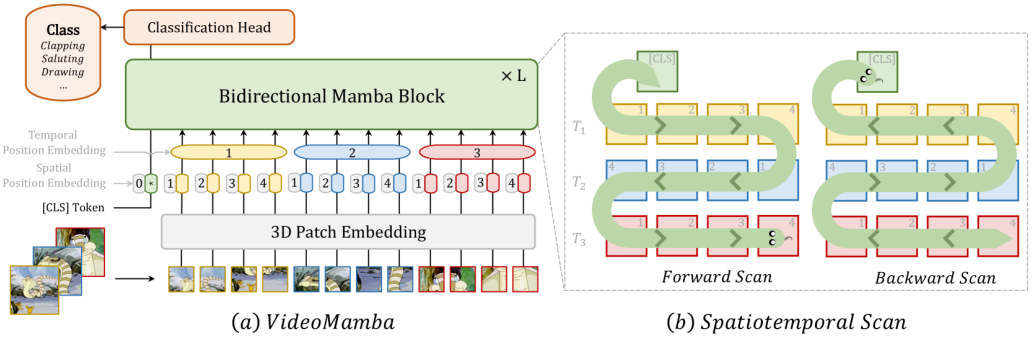
Video classification involves the assignment of categorical labels to video sequences based on the actions and objects depicted in the sequences. In video captioning, the goal is to generate coherent and contextually accurate textual descriptions of video content. This task requires the model to recognize objects and actions within a video and understand the temporal relationships and interactions that occur over time. Traditional approaches to video classification and captioning primarily relied on CNNs to capture spatial features and recurrent neural networks (RNNs) to model temporal dependencies [KTS+14, Has21, MWXZ18]. However, recent advancements have introduced transformer-based models, which offer a more comprehensive approach by simultaneously attending to both spatial and temporal dimensions within video data [ADH+21, LNC+22, YXA+22, LLL+22, WYH+22]. These models leverage the self-attention mechanism [Vas17] to better capture long-range dependencies and complex interactions within video sequences.

In most works, encoder-decoder models have been the preferred approach for classification and captioning tasks. The encoder extracts rich, high-dimensional features from the input video, effectively summarizing the spatial and temporal information into a high-dimensional representation. The decoder leverages this representation to perform specific tasks, such as assigning a class label in video classification or generating a coherent sequence of words in video captioning. The encoders used for the experiments are the pre-trained SVT and Video Mamba backbones, while the decoders are either fully connected neural networks for classification or deep attention-based language decoders for captioning tasks.



(a) *Mamba* (b) *Bidirectional Mamba*

Figure 2: Mamba layers from [LLW+24]



(a) *VideoMamba* (b) *Spatiotemporal Scan*

Figure 3: Video Mamba architecture from [LLW+24]

2.2 Self-supervised Video Transformer

The Self-supervised Video Transformer (SVT) model is a self-supervised transformer-based model pre-trained on the Kinetics-400 data [KCS+17]. The architecture is based on the TimeSformer [BWT21] model that achieves spatiotemporal attention by performing spatial attention across the spatial domain (in each frame of the video) and temporal attention across the temporal domain (across frames). Thus, it is an extension of the original "base" ViT [Dos20] architecture.

The self-supervised training process of the model is carried out through a teacher-student setup for self-distillation [ZSG+19]. Architecturally, the teacher model is a copy of the student model. But, the teacher model is fed only a global sample of the whole video i.e. sampling frame rates are T=8 and T=16 on the video. The student model is fed both the global sample as well as a local sample i.e. frames from a randomly selected sample from the original video. This ensures that the training process is carried out on a multiple-resolution basis, thus improving the model's performance. During one training step, the student model weights are updated via backpropagation while the teacher's weights are updated as an exponential moving average (EMA) of the student weights. The losses consist of two components: $(\mathcal{L}_{gg})$ and $(\mathcal{L}_{lg})$. $(\mathcal{L}_{gg})$ is the loss between the global views of the student and teacher models, to learn the motion correspondences. $(\mathcal{L}_{lg})$ is the loss that compares the local views of the student model with the global views of the teacher model to learn cross-view correspondences. Thus, given f_{g_t} (f_{g_s}) is the teacher (student) model output feature and $f_{l_s}^{(i)}$ are the $K = 8$ extracts of a video from which the local sampling is created, the training loss is:

$$\mathcal{L} = \mathcal{L}_{gg} + \mathcal{L}_{lg} \tag{1}$$

, where

$$\mathcal{L}_{gg} = -f_{g_t} \cdot \log(-f_{g_s}) \tag{2}$$

and

$$\mathcal{L}_{lg} = \sum_{i=1}^{K} -f_{g_t} \cdot \log(-f_{l_s}^{(i)}) \tag{3}$$

The SVT architecture has been shown in Fig. 1.

2.3 Video Mamba

The Video Mamba model [LLW+24] (VM) is a state space model for video-based data. State Space Models (SSMs) are based on continuous systems that map a given input state $x(t) \in \mathbb{R}^L$ to $y(t) \in \mathbb{R}^L$ through a hidden state $h(t) \in \mathbb{R}^N$ as per the following equations:

$$h^{'}(t) = \mathbf{A}h(t) + \mathbf{B}x(t) \tag{4}$$
$$y(t) = \mathbf{C}h(t) \tag{5}$$

where $\mathbf{A} \in \mathbb{R}^{N\times N}$ represents the system's evolution matrix, and $\mathbf{B}, \mathbf{C} \in R^{N\times 1}$ are the projection matrices. This continuous ordinary differential equation (ODE) is approximated through discretization in modern SSMs, such as Mamba. It includes a timescale parameter $\mathbf{\Delta}$ to transform $\mathbf{A}, \mathbf{B}$ into discrete parameters $\bar{\mathbf{A}}$, $\bar{\mathbf{B}}$. The transformation is defined by:

$$\bar{\mathbf{A}} = \exp(\mathbf{\Delta A}) \tag{6}$$
$$\bar{\mathbf{B}} = (\mathbf{\Delta A})^{-1}(\exp(\mathbf{\Delta A}) - \mathbf{I}) \cdot \mathbf{\Delta B} \tag{7}$$
$$h_t = \bar{\mathbf{A}}h_{t-1} + \bar{\mathbf{B}}x_t \tag{8}$$
$$y_t = \mathbf{C}h_t \tag{9}$$

For the Mamba implementation, $\mathbf{B}, \mathbf{C} \in \mathbb{R}^{B\times L\times N}$ and $\mathbf{\Delta} \in \mathbb{R}^{B\times l\times D}$ are directly derived from the input $x \in \mathbb{R}^{B\times L\times D}$, thus allowing for contextual sensitivity and adaptive weight modulation.

The Video Mamba model is built on top of the bidirectional Vision Mamba (B-Mamba) model [ZLZ+24], which has a ViT-like structure. The input video is divided into T frames and each frame is divided into non-overlapping P patches. Each patch is encoded into an embedding through a 3D convolution, followed by the concatenation of a learnable CLS token X_{cls} and positional embeddings. Two types of positional embeddings are used, one for frame order (temporal position) p_t and another for patch order (spatial position) in each frame p_s. Therefore the input X is transformed as

$$X_{\prime} = [X_{cls}, X] + p_s + p_t \tag{10}$$

This is then passed through a bidirectional Mamba block, that contains L SSM layers to perform the spatiotemporal bidirectional operation. The Video Mamba architecture has been shown in Fig. 3.

2.4 Generative Pre-trained Transformer 2 (GPT-2)

A number of models can be used for video captioning, including RNNs, and attention-based networks. With recent success in captioning tasks, the GPT-2 model [RWC+19] is considered as a suitable model for this purpose. The core of GPT-2 consists of multiple stacked transformer blocks, with up to 48 layers. Each transformer block is composed of two main sub-layers: the multi-head self-attention mechanism and a feed-forward neural network.

In addition to the self-attention mechanism, GPT-2 includes positional encoding, which provides the model with information about the order of words in a sequence, giving it sequential processing capabilities. Furthermore, each transformer block incorporates layer normalization and residual connections, which help stabilize training and improve gradient flow through the network.

3 Methods

In this section, the final video classification and captioning pipelines are reviewed that form the basis of all the experiments performed.

3.1 Video Classification Pipeline

The resolution of a video of time length T is limited by its frame rate (f_v) along with the dimensions of each frame. However, for high f_v it is not computationally feasible to input every frame to the model. At the same time, pre-trained SVT and VM encoder models made publicly available have a limited input size d. To fit more frames while being limited by the layer size, we devise a parallel processing framework.

A video is split into $f_v \times T$ frames and a uniform sampling of f_s frame rate is performed such that $d < f_s < f_v \times T$ and d divides f_s. For this work, $f_v = 30$ and f_s are chosen at 1.5. This implies that for each second of video only 1.5 frames are taken instead of the original 30. Further, f_s/d segments of frames are sampled. The model consists of the video encoder as well as the decoder head which in this case is a fully connected neural network. Each set is passed through the model that encodes the information contained in that segment and outputs the action classes present. To obtain a final classification result, all the segment results are pooled by each class using maximum aggregation of the in-

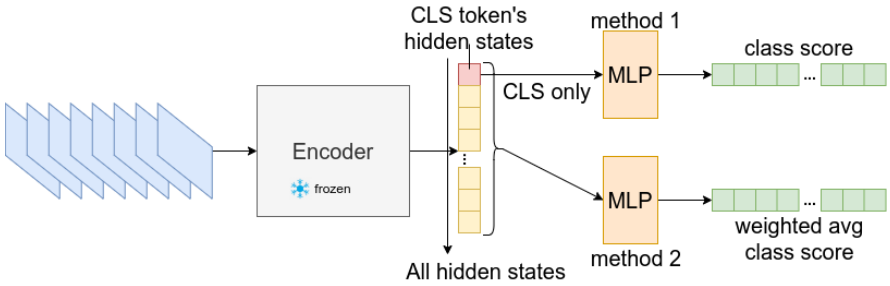


Figure 4: Video Classification pipeline

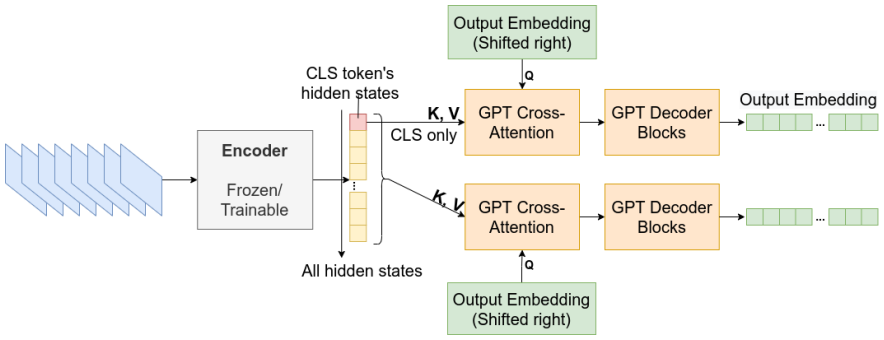


Figure 5: Video Captioning pipeline

dividual prediction scores. Thus, the classification result is an aggregation of the individual classification results across each video segment input to the model. We coin this procedure as the sliding window process. Multiple sliding window strides with overlaps have been experimented with. The pipeline has been visualized in Fig. 4.

3.2 Video Captioning Pipeline

Video Captioning suffers from similar problems and thus a similar approach is followed. However for captioning long-range references are important and only 8-16 frames are not sufficient for capturing all the necessary information. The captions developed from multiple segments would not be sufficient to form a final output. Instead, the video encoder outputs for each clip are stacked together and then passed onto the GPT-2 decoder. This allows the GPT-2 model to find related information across the whole video while still allowing the model to train on the whole video. Additionally, a separate effort is pursued to train the encoder layers that limit the time dimension and allow to inclusion of more frames per clip. Although this makes it more computationally expensive, it is found that a combination of the two approaches leads to the best results. The pipeline has been visualized in Fig. 5.

4 Experiments

In this section, the dataset used for our experiments has been explored. It is followed by a thorough discussion of the training setup for the video classification and captioning tasks.

4.1 Dataset

The Charades dataset [SVW+16] is a comprehensive video dataset focused on daily indoor activities. It comprises 9,850 videos, in which 267 actors were provided with prompts describing specific activities they needed to act out. These interactions were recorded and both class and caption annotations for each video were generated. A consensus approach, involving at least four annotators per video, ensured the accuracy of the labels. The dataset is notable for its scale and diversity, featuring 66,500 classified actions, making it a rich resource for video classification, along with 7,847 captioned videos.

However, performing any task on the Charades

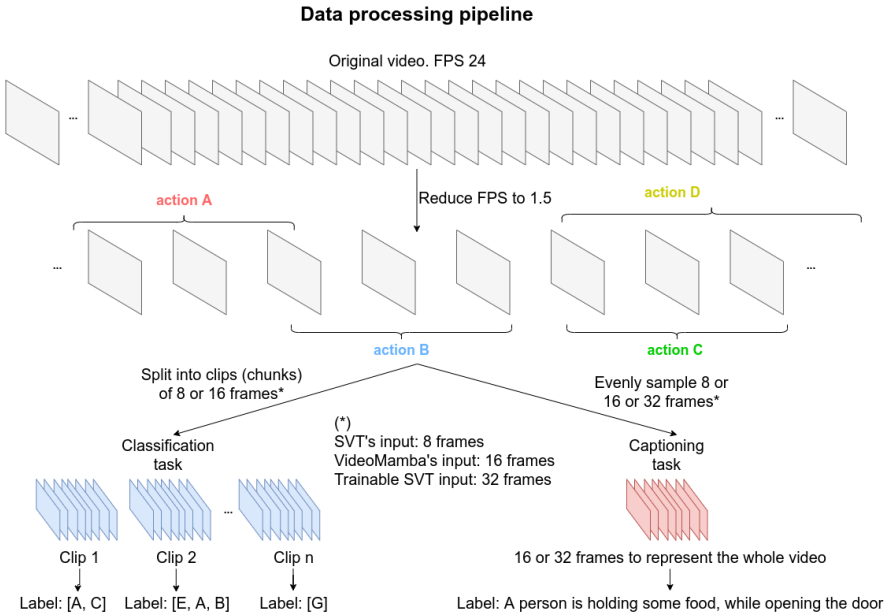


Figure 6: Data Processing pipeline

dataset is non-trivial due to its unique challenges. Each video can contain multiple overlapping actions, necessitating a multi-label classification approach across 157 different action classes. The action annotations are highly unbalanced, with a significant disparity in the relative representation of certain classes with some actions appearing 50 times more frequently than others. Additionally, the temporal annotations of actions are often imprecise, with some extending beyond the video's duration. The video classification and captioning approaches mentioned above are thus taken to overcome the limitations present in the dataset. The full processing pipeline has been visualized through Fig. 6.

4.2 Training setup

4.2.1 Video Classification

The classification models were trained using a multi-layer perceptron (MLP) head with two hidden layers, optimized for each encoder architecture. For the SVT, we used MLP layers of 512 nodes, while for the VM, the same was 1024. Both models utilized the AdamW optimizer, with learning rates fine-tuned to 1e-4 for SVT and 1e-5 for VM to ensure stable training. To address the significant class imbalance present in the Charades dataset, the importance of positive samples is increased by a factor of 2 during training. Our hyperparameter configuration included a dropout rate of 0.7 to prevent overfitting, and we found that excluding LayerNorm improved performance. Image normalization was standardized across both models with a mean of [0.485, 0.456, 0.406] and a standard deviation of [0.229, 0.224, 0.225]. To thoroughly explore the models' capabilities, we conducted extensive experiments comparing the use of only the CLS token versus all token hidden states for classification. Additionally, we investigated various sliding window inference strides to optimize the trade-off between computational efficiency and performance, with strides of 3 seconds and 6 seconds showing the best results. Further experiments were carried out to address the class imbalance issue. We tested different class weighting schemes, including constant weights and dynamic weights based on the ratio of positive to negative samples. These experiments aimed to improve the model's performance on underrepresented classes without sacrificing overall accuracy.

4.2.2 Video Captioning

For the video captioning task, we employed a GPT-2 Small (S) model featuring 12 decoder layers as

our generative head, chosen to balance strong performance with our limited computational resources. The training process utilized the AdamW optimizer with carefully tuned hyperparameters: a learning rate of 2e-4, epsilon set to 1e-6 for numerical stability, and betas of [0.9, 0.999] to control the decay rates of moving averages. We set the batch size to 4, which was the largest possible given our computational constraints while still allowing for effective training. The maximum output sequence length was capped at 128 tokens, which we found sufficient to capture the content of the vast majority of video captions in the Charades dataset. The cross-entropy loss served as our training objective, providing a robust measure for sequence generation tasks.

Our experimental approach was comprehensive, exploring various configurations of encoder and decoder trainability. These ranged from fully frozen encoders with only the generative head being trainable, to intermediate setups with trainable linear projections, to trainable encoders. To enhance the model's ability to capture temporal information, we increased the temporal resolution from the original 8 frames to 16 or 32 frames for the SVT model. This modification necessitated retraining the encoder to accommodate the extended temporal dimension, a process that was computationally demanding but ultimately beneficial for performance. We conducted parallel experiments using both the CLS token exclusively and all hidden states as input to the decoder, aiming to understand the trade-offs between computational efficiency and degree of representation. These experiments aimed to provide valuable insights into the optimal configuration for video captioning on the challenging Charades dataset.

5 Results & Discussion

The results and discussion have been divided into two subsections for video action classification and video captioning for better understanding.

5.1 Video Classification

Our experiments with video classification yielded promising results, demonstrating the effectiveness of both the SVT and Video Mamba models in capturing complex actions in videos. The performance of these models is summarized in Table 1.

Model	No. of Params (M)	mAP
SVT (all)	121.0	19.23
SVT (CLS only)	121.0	25.01
VM (all)	73.6	24.57
VM (CLS only)	73.6	29.82
MViT [FXM$^+$21]	36.4	**46.3**

Table 1: Classification Performance Comparison. CLS only refers to the CLS token strategy used for classification, while all means all hidden states used.

Video Mamba demonstrated superior performance, achieving a mean Average Precision (mAP) of 29.82, compared to SVT's 25.01. This observation is significant since Video Mamba has fewer parameters (73.6 M vs. 121 M for SVT) while processing twice as many input frames. These results suggest that Video Mamba's architecture may be more efficient in capturing temporal dynamics in video data.

It's important to note that while our models show promising results, they still fall short of some state-of-the-art methods benchmarked on the Charades dataset [FXM$^+$21]. This gap can be attributed to our use of frozen encoders, fewer input frames, and smaller architectures compared to fully trainable, larger models used in top-performing approaches.

5.1.1 Hidden State Utilization

Contrary to our initial expectations, using only the CLS token's hidden state outperformed using all hidden states. For Video Mamba, the CLS-only approach achieved an mAP of 29.38, while using all hidden states resulted in an mAP of 24.57. This finding suggests that the CLS token effectively condenses relevant information for classification, possibly due to the nature of the self-supervised pre-training process.

5.1.2 Sliding Window Inference

Implementing overlapping windows during inference led to modest improvements in performance. For SVT, using a stride of 3 seconds increased the mAP from 24.87 to 25.01. Video Mamba showed more substantial gains, with mAP improving from 29.38 to 29.82 using a 3-second stride, and 29.60 with a 6-second stride.

Encoder	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-1 F1	ROUGE-2 F1
SVT-8	0.176	0.075	0.028	0.015	0.177	0.042
SVT-16	0.204	0.061	0.037	0.021	0.229	0.058
SVT-32	0.225	0.106	0.049	0.025	**0.310**	**0.079**
VM	0.233	0.104	0.046	0.023	0.288	0.065
[ZLL+18]	**0.509**	**0.308**	**0.196**	**0.134**	–	–

Table 2: Captioning Performance Comparison. SVT-8 refers to the input temporal dimension.

5.2 Video Captioning

For the video captioning task, we evaluated our models using standard metrics including BLEU and ROUGE scores. Table 2 presents the results for our best-performing models.

Both SVT and Video Mamba achieved BLEU-1 scores above 0.22, indicating a reasonable ability to generate accurate captions. The SVT model slightly outperformed Video Mamba in most metrics, particularly in ROUGE scores, suggesting it may produce more coherent and comprehensive captions.

5.2.1 Frame Count

Increasing the temporal dimension for SVT led to modest improvements in caption quality, highlighting the importance of temporal context in video understanding

5.2.2 Hidden State Utilization

In contrast to classification tasks, leveraging all hidden states proved more effective for captioning. This strategy allows the GPT-2 decoder to utilize context from each frame, resulting in more detailed captions. This underscores the distinct requirements between captioning and classification.

5.2.3 Model Trainability

The top-performing model was achieved using fully trainable encoders and decoders, highlighting the advantages of end-to-end fine-tuning for this task. While our models demonstrate the ability to generate relevant captions, there remains a significant gap between our results and those reported in some previous works [ZLL+18]. This discrepancy may be attributed to our use of fewer input frames and smaller model architectures.

6 Conclusion

In this paper, we investigated the application of transformer-based models for video classification and captioning on the complex Charades dataset. Our experiments demonstrate the effectiveness of these models in capturing intricate human interactions and daily life activities, even with limited computation and smaller architectures. For video classification, Video Mamba outperformed SVT, achieving a higher mean Average Precision while processing a higher temporal dimension with fewer parameters. In the video captioning task, both SVT and Video Mamba generated relevant captions, with SVT slightly outperforming Video Mamba in most metrics.

Increasing the temporal dimension and leveraging all hidden states improved caption quality, emphasizing the importance of temporal context and frame-level information for this task. While our models demonstrate promising results, there remains room for improvement when compared to state-of-the-art methods. Future work could focus on scaling up the models, incorporating higher temporal dimensions, and exploring more sophisticated training techniques to further bridge this performance gap. Additionally, establishing official benchmarks for video captioning on the Charades dataset would facilitate more direct comparisons. Overall, our research contributes to the growing body of work on transformer-based models for video understanding tasks, showcasing their potential in handling complex video data and reducing reliance on large labeled datasets.

References

[ADH+21] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of*

the IEEE/CVF international conference on computer vision, pages 6836–6846, 2021.

[BWT21] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021.

[Dos20] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[FXM+21] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6824–6835, 2021.

[Has21] Ehtesham Hassan. Learning video actions in two stream recurrent neural network. *Pattern Recognition Letters*, 151:200–208, 2021.

[KCS+17] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[KTS+14] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[LLL+22] Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, Yumao Lu, and Lijuan Wang. Swinbert: End-to-end transformers with sparse attention for video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17949–17958, 2022.

[LLW+24] Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. Videomamba: State space model for efficient video understanding. *arXiv preprint arXiv:2403.06977*, 2024.

[LNC+22] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3202–3211, 2022.

[MWXZ18] Feng Mao, Xiang Wu, Hui Xue, and Rong Zhang. Hierarchical video frame sequence representation with deep convolutional graph network. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018.

[RNK+22] Kanchana Ranasinghe, Muzammal Naseer, Salman Khan, Fahad Shahbaz Khan, and Michael S Ryoo. Self-supervised video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2874–2884, 2022.

[RWC+19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[SVW+16] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 510–526. Springer, 2016.

[Vas17] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[WYH+22] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022.

[YXA+22] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3333–3343, 2022.

[ZLL+18] Bin Zhao, Xuelong Li, Xiaoqiang Lu, et al. Video captioning with tube features. In *IJCAI*, pages 1177–1183, 2018.

[ZLZ+24] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024.

[ZSG+19] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3713–3722, 2019.