Two-Stream Transformer Architecture for Long Form Video Understanding

Edward Fish
edward.fish@surrey.ac.uk

Jon Weinbren
j.weinbren@surrey.ac.uk

Andrew Gilbert
a.gilbert@surrey.ac.uk

The University of Surrey
Guildford, UK

Abstract

Pure vision transformer architectures are highly effective for short video classification and action recognition tasks. However, due to the quadratic complexity of self attention and lack of inductive bias, transformers are resource intensive and suffer from data inefficiencies. Long form video understanding tasks amplify data and memory efficiency problems in transformers making current approaches unfeasible to implement on data or memory restricted domains. This paper introduces an efficient Spatio-Temporal Attention Network (STAN) which uses a two-stream transformer architecture to model dependencies between static *image* features and temporal *contextual* features. Our proposed approach can classify videos up to two minutes in length on a single GPU, is data efficient, and achieves SOTA performance on several long video understanding tasks.

1 Introduction

Long form video understanding (LVU) is a sub-domain of video recognition concerned with understanding contextual information across contiguous shots which can contain multiple locations, scenes, interactions, and actions. While a blurb can give a snapshot of a book, stories are enriched by the development of characters and their interactions with objects, people, and locations. The same applies in video recognition where understanding the context of individual moments in relation to a whole video can provide valuable information for tasks such as classification, speaker recognition, character understanding, and video retrieval. However, current video recognition methods and datasets have tended to focus on short videos [7, 16, 19, 27, 29, 32, 37, 44, 55], effectively aggregating convolutional image features via late fusion and inflation [16, 32, 33, 44, 52], or extending attention based image classification methods to the tasks of action recognition, object tracking, and segmentation [4, 21, 26, 36, 43, 55].

Intuitively, an image can provide a good summary of a moment in time, and so it is logical that these methods perform well at classifying short videos from only a few frames [51]; however, many video classification tasks require, or can be improved with, long-term temporal reasoning [56]. The temporal fusion of features via recurrent networks [12, 45] provides one solution by aggregating frames over longer sequences of frames, but these architectures suffer from computational inefficiencies due to their recursive design while 3D convolutional
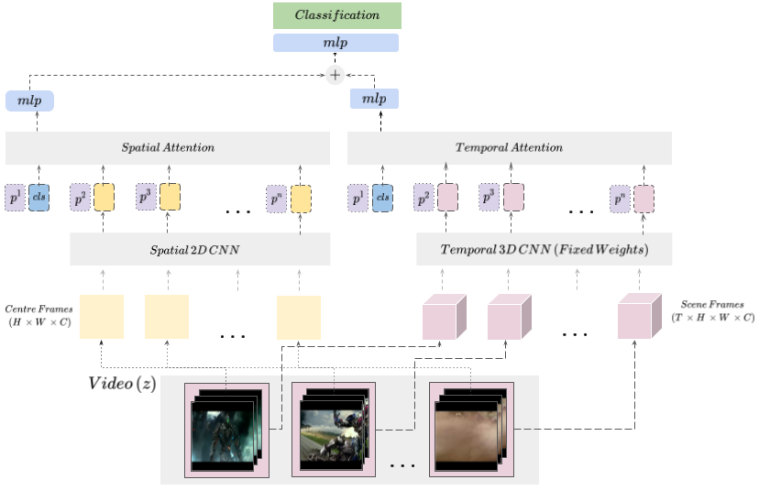
Figure 1: An overview of our approach, STAN. We encode video scenes into two feature representations using a two-stream spatio-temporal convolutional network. A transformer encoder is then used to model temporal dependencies between the tokens via an additional classification token randomly initialised. The proposed method allows us to model long term dependencies between individual frames of long videos which feature multiple actions and environments.

approaches also do not scale effectively, primarily due to their high computational footprint and the linear growth of the receptive field in the temporal dimension.

More recently, transformers [11, 14] have been adapted to the video domain [2, 21, 48], achieving SOTA results on multiple short video tasks. But due to the lack of inductive bias in transformer architectures, it has been observed that video transformers are not as data efficient as their CNN counterparts [49], requiring more data and thus more time to train. Furthermore, it has been shown that transformer image networks trained on natural videos work just as effectively when inputs are shuffled [15], demonstrating weak temporal bias. This issue is compounded when addressing LVU tasks, as we encounter larger datasets with fewer opportunities for parameter updates during training caused by a small target to data ratio. The following question then arises.

How can we leverage inductive bias from image CNN's to make video transformer networks more data and memory efficient for long form video understanding tasks?

Our solution is a two-stream Spatio-Temporal Attention Network (STAN) with which we gain data and computational efficiency by introducing inductive bias via convolution. Existing methods for image classification with transformers such as [14] split images into 16×16 pixel regions encoded with a positional embedding to introduce permutation and translation invariance. This method has been extended to video by expanding these regions temporally to create 3D tokens with 2D positional embeddings [2]. Inspired by work in two-stream slow-fast convolutional video networks for short video classification [17], our approach replaces this tokenisation method with both *image* spatial and *context* temporal scene features extracted from pre-trained convolutional neural networks. The method, as shown in Fig 1 is 9-16x more memory efficient than current video transformer methods such as [41] as we re-purpose the positional encoding to model temporal dependencies between scenes, rather

than attention between pixel regions. As the transformer network requires quadratic computation and linear memory with each additional token, minimizing the number of tokens, layers, and heads of the transformer enables us to re-purpose that memory to process more frames, while benefiting from inductive bias via the CNN spatial stream. Our method enables long form video classification on a single consumer grade GPU, achieving SOTA classification accuracy on videos up to two minutes in length.

2 Related Work

Early video classification works focused on non-temporal aggregation, which included clustering [31, 39, 43] or fusing [18] spatial features obtained from convolutional neural networks. Since a short video will share a similar distribution of pixels over concurrent frames, these networks perform well for video classification and object detection tasks. Naturally, fusing these output features using RNN's [3, 8, 45, 57] such as an LSTM [12, 61] improves performance by introducing temporal information. However, as discussed in [14], these model architectures align with steps in computational time, and are inefficient at longer sequence lengths as memory constraints limit batching across sequences.

Later works explored extending convolution to video, inflating image CNN's via a temporal channel [6, 16, 32, 53, 44] and using two stream convolutional networks to aggregate spatial and temporal information introduced via optical flow [47], or from various sampling and aggregation intervals [42, 51, 58]. 3D Convolutional Neural Networks (CNN's) are highly effective at video classification, object detection, and action recognition tasks but are computationally intensive to train. For example, in [50] the authors process just 16 frames at the cost of 40 GFLOPS per single pass, making the approach infeasible for sequences longer than a few seconds. To address computational overhead in training convolutional video networks, in [17] the authors present an efficient and high performance CNN for video classification, which utilises a fast and slow temporal sampling stream. The slow stream can process higher resolution images and extract key spatial information over a few key frames, while the fast stream maps low resolution frames to infer temporal information. This method is highly effective at video and action classification on short 10 second clips at a low computational cost.

Transformers [5, 11, 14, 53] use self-attention to model dependencies between inputs and have recently been shown to work effectively for video classification tasks when implemented with temporal positional information [4, 21, 26, 36, 48, 55]. Unlike convolutional methods, they lack inductive bias and, as such, take longer to model dependencies between neighbouring pixel regions which extends training time and harms data efficiency [49] thus making them innapropriate for LVU tasks. Introducing inductive bias via convolution, shifting windows, and gaussian bias to transformers has shown to be effective in the image domain [23, 34, 54, 59, 60], and hybrid networks for video have also been proposed in [9, 22, 41], but only on short segments of video. Finally, LVU networks have been explored in [56, 53] where the authors leverage short-term detection and tracking to form instance representations or use LSTM's for the task of long form video-question answering respectively.

3 Method

Overview We aim to classify long videos by splitting them into discrete scenes and extracting spatial and temporal representations for temporal fusion via a two-stream transformer encoder. To do so, we first analyse the video for changes in average frame intensity/brightness using a running average over RGB video channels. We use these time stamps as scene segmentation

points and uniformly sample 12 frames from each scene segment using a 3D CNN encoder to generate a low-resolution temporal feature tokens for each scene. We also extract the central frame of each scene at a higher resolution and use a 2D CNN to obtain a spatial feature token. While this sampling method results in variable length scenes, the segmentation threshold is met only when there is a substantial shift in the rgb values. This means that scenes with more dynamic content are split, and therefore sampled more densley than scenes that contain longer periods of static content. The spatial and temporal scene tokens are encoded with a positional embedding and temporally aggregated using a two-stream transformer encoder. For classification, we randomly initialise an additional token prepended to the spatial and temporal sequence of embedding tokens, which learns to model the temporal inter-dependency of the individual scenes. An overview of the sampling methodology is shown in Fig 2.
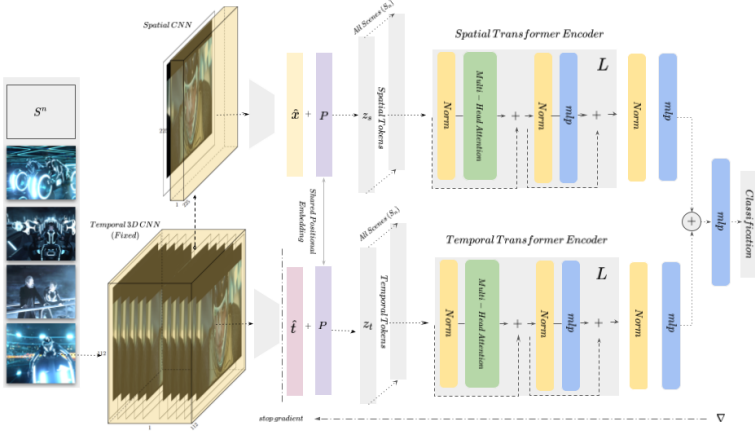

Figure 2: For each scene s in the video v we obtain a temporal ($\hat{t}$) and spatial ($\hat{x}$) feature embedding using 2D and 3D convolutional neural networks. A shared positional embedding, p, is added to every scene embedding to generate the spatial tokens, z_s, and the temporal tokens, z_t. A two stream spatio-temporal transformer with two layers (l), learns the dependency between the sequence of spatial tokens and the temporal tokens. Following normalisation and a linear projection, the features are fused for classification. In practice, we only fuse and classify the prepended CLS token as discussed in Section 3. When training the STAN-Small model, we do not back-propagate through the 3D CNN represented here by the stop-gradient line. For STAN-Large we continue to fine-tune both convolutional encoders.

Temporal Encoding Token Given a uniformly sampled set of 12 video frames from a scene defined here as $t \in \mathbb{R}^{12 \times H \times W \times C}$ we implement a R(2+1)d Video ResNet encoder [51] pre-trained on the Kinetics400 Dataset [6], defined as $g(t)$, to encode each set of frames into a single feature embedding token $\hat{t}$ which represents the projected temporal features of the scene s_i from a set of scenes s. We obtain an embedding for each scene in the video using the above method to generate a set z_t which represents the set of all temporal embedding vectors for the scene $s_i \in s$.

$$z_t = [z_{cls}, \hat{t_0}, \hat{t_1}, ..., \hat{t_n}] + p \qquad (1)$$

Where z_{cls} is a randomly initialized vector in the same dimension as $\hat{t_i}$ and $p_i \in \mathcal{PE}$ is a positional embedding added to the sequence as described in [53] as,

$$\mathcal{PE}_{pos,2i} = sin(\frac{pos}{10000^{2i/d_{model}}})$$

$$\mathcal{PE}_{pos,2i+1} = cos(\frac{pos}{10000^{2i/d_{model}}}) \tag{2}$$

Where d_{model} is a common dimension for both the spatial and temporal tokens. In [14, 22, 62], positional information is used to infer the relationship between cropped regions in an individual frame. As we have introduced inductive bias via convolution, we do not need to model the position of pixel regions and instead extend the positional embedding method to infer the position of each short temporal sequence. We argue that this is logical for longer videos that feature a narrative composed of multiple dynamic actions and environments such as those found in movies. In section 4 we show that this positional information benefits the model's performance on such a task.

Spatial Encoding Token To obtain a spatial encoding token, we perform a linear projection of the high resolution frame x, which is sampled from the center of s_i, to a spatial embedding token $\hat{x}$ using a ResNet18[24] encoder model pretrained on the ImageNet Dataset [10] so that $\hat{x} = h(x)$. As in the case of the temporal token, we form a sequence of spatial embedding tokens z_x from spatial feature embeddings obtained via $h(.)$ so that $z_x = \{\hat{x_0}, \hat{x_1}, ..., \hat{x_n}\}$ where n is the total number of scenes in the video z. In Eq 3 z_x^{cls} is the randomly initialized spatial cls token used for classification.

$$z_s = [z_x^{cls}, \hat{x_0}, \hat{x_1}, ..., \hat{x_n}] + p \tag{3}$$

The positional embedding vector p in Eq 3 is the same as defined in Eq 1 and as such the positional information is shared between the spatial and temporal tokens. This ensures that positional information is consistent between the spatial and temporal streams during fusion. We now define a spatial and temporal transformer for temporal aggregation of the input tokens and then discuss techniques to fuse the output classification from both streams.

Spatial and Temporal Transformer Encoders For the spatial and temporal transformer encoder architecture, we implement the transformer model introduced in [53] originally designed for natural language processing. As described in [53] we generate Query (Q), Key (K), and Value (V) matrices from both the spatial and temporal embedding representations where each row in the matrix represents a corresponding scene, with the first row representing our classification token. A matrix of outputs is computed as,

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}V}) \tag{4}$$

The output matrix $A(Q, K, V)$ is summed with the input embedding matrix via a residual connection and normalised using Layer Normalisation. Finally, the output is summed with a second residual connection before a classification MLP head. In practice, we only apply the final classification MLP to the temporal and spatial transformer classification tokens which learn a representation of the input via self-attention. As such, we can discard the other rows of the output matrix and only backpropagate via the classification MLP. We use Multi-Headed-Self-Attention to model further representations in spatial and temporal domains by replicating the attention mechanism in Eq 4 and concatenating the heads. This process is described in detail in [53]. In [9, 22, 62] the authors use a linear layer for classification based on $z_{cls} \in \mathbb{R}^d$.

In our work, we linearly project z_{cls} to $\mathbb{R}^d_{model}$ for both the spatial and temporal features so we can experiment with several fusion methods, which we will describe next.

Fusion and Classification For fusion, we normalise the output embeddings and project to a common dimension with a linear layer before classification via a three-layer MLP separated by a gated non-linearity. The loss function can be defined as a binary cross entropy loss between the targets y and the scaled sum of the feature embeddings,

$$\mathcal{L}^{Fusion} = L_{BCE}(f(Norm([q(z_s^{cls}) + \lambda q(z_t^{cls})]), y) \tag{5}$$

Where $\lambda < 1.0$ acts as a hyper-parameter to scale the influence of the temporal network in generating the output logits and set to 0.6. $q(.)$ Is an MLP with one hidden layer and weights shared for both the temporal and spatial streams, while the function $f(.)$ represents the final MLP used for classification after fusion. To improve training time we use transfer learning and pre-train the 2D CNN on ImageNet [10] and the 3D CNN on Kinetics400 [6]. We present both a large, and small version of the model. For the **STAN-Small**, we do not update the parameters in the temporal stream layers but back-propagate through the two-stream transformer to the 2D CNN. This reduces the number of trainable parameters by 48 million, making the whole network trainable on a single Nvidia RTX5000 GPU with 16GB of memory. The larger model, **STAN-Large**, which back-propagates via both streams, improves performance by 11% but requires 92.5 million trainable parameters compared with 45 million for STAN-Small. In Table 1 we experiment with a number of additional methods for fusion including collaborative gating [35] and distillation [25]. A full experimental analysis follows in the next section.

In practice, the method could be extended to any arbitrary length of tokens, by both increasing and decreasing the frame sampling frequency for variable length scenes used by the temporal stream, or extending the number of tokens.

Method	CNN Feature Extractor	Frames per Scene	Aggregation Method	$\overline{mAP}$
ResNet [24]	ResNet18	1	Avg Pool	0.434
SqueezeExcite [28]	SE-ResNet	1	Avg Pool	0.544
I3D [6]	I3D	12	Avg Pool	0.487
Collaborative Gating [35]	ResNet18	12	Gated Unit	0.4723
S(TPN) [52]	ResNet18	12	Temporal Pyramid	0.492
Fine-Grained Semantic [18]	Multi-Modal	16	Concatenation	0.583
LSTM [12]	ResNet18	1	LSTM	0.596
Distillation [25, 49]	ResNet18 + R2+1D	12	Transformer	0.601
STAN-Small	ResNet18 + R2+1D	12	Transformer	**0.640**
STAN-Large	ResNet18 + R2+1D	12	Transformer	**0.750**

Table 1: Comparison of our proposed approach with existing methods for video classification using CNN feature extractors and evaluated on the MMX-Trailer-20 Dataset. We implement several feature extraction and aggregation methods to evaluate their effectiveness for the long video classification task.

4   Results

We evaluate the proposed method on several long video classification tasks. As discussed in [56] video datasets have typically focused on short video tasks, therefore following [56]

we define long videos as videos which feature more than three shots or scenes, are longer than a minute in length, and in which classification relies on, or is improved by, contextual understanding of the relationship between the content of shots and their order. To evaluate our method, we use both the **MMX-Trailer-20 Dataset (MMX)** [18], and the **Long Video Understanding Benchmark (LVU)** [56].

MMX Dataset The MMX-Trailer-20 Dataset [18] is a multi-modal movie trailer dataset with 7555 movie trailers spanning many scenes, locations, actions, and narratives. The MMX-Trailer dataset features movie trailers from 20 genres with six genre labels per video from $1900 - 2021$ with an average length of two minutes. The training set is 6047 videos, with the validation and test sets both at 754 samples. The dataset consists of 37 million frames in total.

LVU Benchmark The Long Video Understanding Benchmark [56] contains 30K videos with an average length of 120 seconds. The benchmark comprises of content understanding, user engagement prediction, and movie metadata prediction tasks, demonstrating various requirements for long temporal modelling. We test our approach on the content understanding tasks including character relationship identification, speaking style, and scene recognition.

Method	Relation	Speaking	Scene
R101-SlowFast+NL [17]	52.4	35.8	54.7
VideoBERT [48]	52.8	37.9	54.9
Object Transformer [56]	53.1	39.4	56.9
STAN-Small (ours)	**54.2**	**40.6**	**57.3**
STAN-Large (ours)	**56.25**	**41.41**	**58.33**

Table 2: Accuracy of our approach on long video understanding tasks using the Long Video Understanding Dataset. The reader can find references and further details in [56]. We outperform current approaches for classifying conversation (speaking), character relationships (relation), and locations of scenes (scene).

Compared Methods To demonstrate the effectiveness of the two-stream network, we implement several existing methods for video classification using the MMX-Trailer-20 dataset. They include extracting convolutional image features from a ResNet18 [24] for each scene and simply using average pooling for classification. We also implement this same method with features obtained from a two-stream Inflated 3D Convolutional Neural Network [32] and a SENet [28]. We also show results for comparative temporal networks, including a Temporal Pyramid Network, which extracts multiple level features from a CNN to model dynamic temporal movements in videos, and a vanilla LSTM [12] to model the temporal relationship between the feature representations from an 18 layer ResNet. More complex temporal aggregation strategies are also explored, such as the Fine-Grained Semantic [18] architecture which uses concatenation to aggregate multi-modal features as well as the effectiveness of earlier audio-visual works such as audio-visual classification using support vector machines [30] and audio VGG network as described in [1]. We also explore distillation as described in [49], using the temporal transformer encoder as a teacher network to the spatial transformer. For the LVU Benchmark task we compare our method with those presented in the paper including videoBERT [48], R101-SlowFast [17] and Object Transformer [56].

Metrics As outlined in [18] the MMX dataset is imbalanced and as such we follow other works [13, 38, 40] and use Mean Average Precision *mAP* to evaluate the effectiveness of our classifier. To calculate the *mAP*, we average the area under the precision-recall curve per genre, weighting instances according to the class frequencies. We also show weighted

Precision (P_w), weighted Recall (R_w), and weighted F1-Score ($F1_w$) in Table 3. With all metrics, a higher value demonstrates improved accuracy. For LVU we compute the standard error averaged over five runs as proposed in the original paper [56] where higher values represent improved performance.

Evaluation First, in Table 1 we evaluate our approach against existing methods for video classification using the MMX-Trailer-20 Dataset. We demonstrate that our method outperforms pooling of convolutional spatial features by 10%. Secondly, we show our method improves performance on scene level features extracted via an inflated 3D CNN [16] by 11%. Third, we compare our approach with other methods for combining convolutional features temporally, including an LSTM [12], S(TPN) [64], concatenation [18], and collaborative gating [55]. Finally, we show results for an alternative method for two-stream aggregation using a distillation network as described in [49]. We outperform all existing methods for the genre classification task.

Model	Actn	Advnt	Animtn	Bio	Cmdy	Crme	Doc	Drma	Famly	Fntsy	Hstry	Hrror	Mystry	Music	SciFi	Sprt	Shrt	Thrll	War	$F1_w$	mAP	P_w	R_w
Support	130	197	46	13	224	102	87	267	117	115	44	104	41	86	107	30	45	12	21	-	-	-	-
Random	0.29	0.41	0.11	0.03	0.46	0.24	0.21	0.52	0.27	0.26	0.11	0.24	0.1	0.2	0.25	0.08	0.11	0.03	0.05	0.318	0.134	0.19	1
ResNet [■]	0.43	0.55	0.74	0	0.49	0.38	0.63	0.55	0.51	0.28	0.24	0.42	0.3	0.28	0.41	0.22	0.19	0.11	0.33	0.434	0.489	0.437	0.48
VGG-Audio [■]	0.47	0.51	0.40	0.10	0.61	0.38	0.58	0.55	0.51	0.37	0.11	0.34	0.39	0.30	0.35	0.16	0.15	0.13	0.12	0.454	0.449	0.400	0.537
I3D [■]	0.5	0.59	0.74	0	0.62	0.33	0.63	0.56	0.55	0.36	0.2	0.38	0.45	0.24	0.37	0.23	0.14	0.10	0.13	0.463	0.487	0.448	0.494
SqueezeExcite [■]	0.48	0.63	0.79	0.12	0.65	0.41	0.60	0.59	0.55	0.42	0.25	0.47	0.42	0.29	0.50	0.34	0.19	0.12	0.31	0.516	0.554	0.493	0.572
Naive Concat [■]	0.56	0.61	0.64	0.09	0.64	0.35	0.69	0.60	0.58	0.39	0.19	0.49	0.45	0.21	0.48	0.39	**0.28**	0.27	0.41	0.525	0.497	0.522	0.551
Fine-Grained Semantic [■]	0.62	**0.69**	0.71	0.11	**0.71**	0.53	**0.73**	0.62	0.51	0.34	0.56	**0.60**	**0.45**	0.50	**0.64**	**0.30**	0.11	0.13	**0.55**	0.597	0.583	0.554	0.697
STAN-Small	**0.71**	0.68	**0.92**	**0.21**	0.61	**0.65**	0.62	**0.69**	**0.86**	0.49	**0.46**	0.58	0.43	0.39	0.53	0.13	0.20	**0.85**	0.50	**0.65**	**0.64**	**0.62**	**0.73**

Table 3: Genre classification performance for each genre on the MMX-Trailer-20 dataset. We observe high performance gains on genres in which temporal information can be considered an important classifier such as Action +9 and Animation +11. We also observe that other network architectures perform very poorly on classification of the genre thriller while we improve accuracy by +58. We conclude that long-term temporal modelling performs well on this task as the content is difficult to classify when features are presented in isolation.

In Table 2 we show that we achieve SOTA results for three long-form video understanding tasks on the LVU Benchmark outperforming other methods which also utilise transformer architectures such as VideoBERT [48] and [56] which utilises pre-trained CNN backbones plus self-supervised masked pre-training. Our architecture performs particularly well on the relationship identification task ($+\textbf{3.1}$) despite having no prior knowledge of the domain in pre-training.

Ablation Experiments In Table 4 we perform several ablation studies to assess the impact of the spatial and temporal features and the data efficiency of the model. To measure the data efficiency, we randomly sampled 2000 trailers from the MMX-Trailer-20 training partition, with the entire test partition of 754. Table 4 shows that the model still achieves SOTA performance on the MMX-Trailer-Dataset despite being trained on only a quarter of the samples demonstrating high data efficiency. We infer that data efficiency is improved by using convolutional encoding as the transformer network only needs to map self attention between the scene feature tokens rather than pixel localities. Furthermore, the proposed network architecture can infer translation in-variance within the convolutional encoding.

In Table 4 we also provide further results for models that only use either the spatial or temporal convolutional encoder to assess the impact of propagating gradients through the convolutional encoders. We observe that back-propagating through the spatial convolutional encoder provides the most significant performance gain with the most negligible effect on the number of trainable parameters. Training both the CNN encoders end to end (STAN-Large) is the most effective method for achieving high accuracy. Still, it comes at a cost, increasing the number of trainable parameters by 48 million. We find that using just the temporal transformer

encoder and introducing a stop gradient before the convolutional feature extractor performs well but is improved with spatial features for 28 million additional trainable parameters. We also show genre specific results in Table 3.

Method	Samples	Parameters (Millions)	mAP
Spatial with backprop	2000	28	0.5903
Temporal with backprop	2000	48	0.6221
Spatial no backprop	2000	16.5	0.4024
Temporal no backprop	2000	16.5	0.59
Distillation Network	2000	44.5	0.6005
Gated Fusion	2000	45	0.4728
STAN-Small	2000	45	0.6151
STAN-Small	6047	45	0.6401
STAN-Large	6047	92.5	0.7506

Table 4: Ablation experiments assess the network quality under a constrained data training protocol. Each model is trained using only 2000 samples, while the test length remained consistent at 754 samples. The network continues to outperform existing methods with fewer data. We also show results for individual spatial and temporal streams with back-propagation.
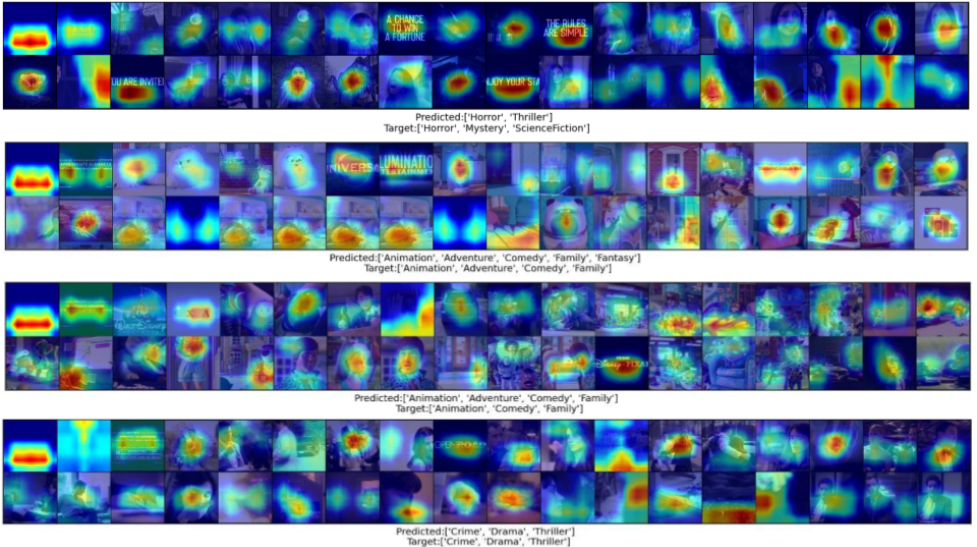

Figure 3: Class activation maps of randomly selected samples from the MMX-Trailer-20 test partition. Each sub-figure represents a series of input tokens from the spatial encoder $h(\hat{x})$. The class activation maps show pixel regions in red to blue, with red regions representing high activation and, therefore, greater contribution to the predicted class labels.

Qualitative Results In Fig 3 we show class activation maps of the input 2D CNN encoder, shown in Fig 2 as $h()$ and the predicted genres for a given input video. We observe that the model predicts the labels correctly and learns cohesive features for the input tokens. For example in Fig 3 we see that the Family genre is predicted when we have a strong activation

on animals faces. For the Horror classification, spatial features include people screaming and dark scenes. We also notice that text is an important feature for the classification task, acting as a strong temporal marker in the sequence of tokens. Class activation maps are obtained via Grad-CAM[46] using the code provided by [20]. We refer the reader to the appendix for further qualitative results and ablation experiments.

5 Conclusions

We present a data and memory efficient spatio-temporal attention network for long video classification which combines the advantages of convolutional inductive bias with the computational advantages of transformer networks for the task of long video classification. We show that by using static *image* and temporal *context* convolutional tokens we are able to create a data efficient architecture capable of classifying videos of up to two minutes in length and trained on a single GPU while achieving SOTA results on a range of long-form video understanding tasks.

References

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[2] 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*, 2021.

[3] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. Sequential deep learning for human action recognition. In *International workshop on human behavior understanding*. Springer, 2011.

[4] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021.

[5] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[7] Paola Cascante-Bonilla, Kalpathy Sitaraman, Mengjia Luo, and Vicente Ordonez. Movie-escope: Large-scale analysis of movies using multiple modalities. *arXiv preprint arXiv:1908.03180*, 2019.

[8] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[9] Stéphane d'Ascoli, Hugo Touvron, Matthew Leavitt, Ari Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. *arXiv preprint arXiv:2103.10697*, 2021.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR-09*, 2009.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[12] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Sub-hashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[13] Jianfeng Dong, Xirong Li, and Cees GM Snoek. Word2visualvec: Image and video to sentence matching by visual feature prediction. *arXiv preprint arXiv:1604.06838*, 2016.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[15] 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*, 2021.

[16] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[17] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2019.

[18] Edward Fish, Jon Weinbren, and Andrew Gilbert. Rethinking genre classification with fine grained semantic clustering. In *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021.

[19] David F Fouhey, Wei-cheng Kuo, Alexei A Efros, and Jitendra Malik. From lifestyle vlogs to everyday interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[20] Jacob Gildenblat and contributors. Pytorch library for cam methods. https://github.com/jacobgil/pytorch-grad-cam, 2021.

[21] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. *arXiv preprint arXiv:2106.02036*, 2021.

[22] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. *CoRR*. URL http://arxiv.org/abs/1812.02707.

[23] Jianyuan Guo, Kai Han, Han Wu, Chang Xu, Yehui Tang, Chunjing Xu, and Yunhe Wang. Cmt: Convolutional neural networks meet vision transformers. *arXiv preprint arXiv:2107.06263*, 2021.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[25] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[26] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv/1912.12180*, 2019.

[27] Chiori Hori, Takaaki Hori, Teng-Yok Lee, Ziming Zhang, Bret Harsham, John R Hershey, Tim K Marks, and Kazuhiko Sumi. Attention-based multimodal fusion for video description. In *Proceedings of the IEEE international conference on computer vision*, pages 4193–4202, 2017.

[28] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[29] Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding. In *European Conference on Computer Vision*. Springer, 2020.

[30] Yin-Fu Huang and Shih-Hao Wang. Movie genre classification using svm with audio and video features. In *International Conference on Active Media Technology*. Springer, 2012.

[31] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *2010 IEEE computer society conference on computer vision and pattern recognition*, 2010.

[32] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2012.

[33] 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*, 2014.

[34] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7083–7093, 2019.

[35] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. Use what you have: Video retrieval using representations from collaborative experts. *BMVC-19*, 2019.

[36] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*, 2019.

[37] Marcin Marszalek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009.

[38] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with Context Gating for video classification, jun 2017.

[39] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*, 2017.

[40] Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze, and Amit K Roy-Chowdhury. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, 2018.

[41] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. *arXiv preprint arXiv:2102.00719*, 2021.

[42] Junting Pan, Siyu Chen, Mike Zheng Shou, Yu Liu, Jing Shao, and Hongsheng Li. Actor-context-actor relation network for spatio-temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[43] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *2007 IEEE conference on computer vision and pattern recognition*, 2007.

[44] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *proceedings of the IEEE International Conference on Computer Vision*, 2017.

[45] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[46] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 2017.

[47] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014.

[48] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7464–7473, 2019.

[49] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablay-rolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*. PMLR, 2021.

[50] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, 2015.

[51] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018.

[52] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

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

[54] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiao-gang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[55] Xiaofang Wang, Xuehan Xiong, Maxim Neumann, AJ Piergiovanni, Michael S Ryoo, Anelia Angelova, Kris M Kitani, and Wei Hua. Attentionnas: Spatiotemporal attention cell search for video classification. In *European Conference on Computer Vision*, pages 449–465. Springer, 2020.

[56] Chao-Yuan Wu and Philipp Krahenbuhl. Towards long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1884–1894, 2021.

[57] Zuxuan Wu, Xi Wang, Yu-Gang Jiang, Hao Ye, and Xiangyang Xue. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 461–470, 2015.

[58] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 305–321, 2018.

[59] Baosong Yang, Zhaopeng Tu, Derek F Wong, Fandong Meng, Lidia S Chao, and Tong Zhang. Modeling localness for self-attention networks. *arXiv preprint arXiv:1810.10182*, 2018.

[60] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021.

[61] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[62] Yanyi Zhang, Xinyu Li, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. Vidtr: Video transformer without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13577–13587, 2021.

[63] Zhou Zhao, Zhu Zhang, Shuwen Xiao, Zhenxin Xiao, Xiaohui Yan, Jun Yu, Deng Cai, and Fei Wu. Long-form video question answering via dynamic hierarchical reinforced networks. *IEEE Transactions on Image Processing*, 28(12):5939–5952, 2019.

[64] Zhenxing Zheng, Gaoyun An, Dapeng Wu, and Qiuqi Ruan. Spatial-temporal pyramid based convolutional neural network for action recognition. *Neurocomputing*, 358:446–455, 2019.

[65] Howard Zhou, Tucker Hermans, Asmita V Karandikar, and James M Rehg. Movie genre classification via scene categorization. In *Proceedings of the 18th ACM international conference on Multimedia*, 2010.