Computer Systems
Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2023.043245

ARTICLE

TEAM: Transformer Encoder Attention Module for Video Classification

Hae Sung Park[1] **and Yong Suk Choi**[2,*]

[1]Department of Artificial Intelligence, Hanyang University, Seoul, Korea

[2]Department of Computer Science, Hanyang University, Seoul, Korea

*Corresponding Author: Yong Suk Choi. Email: cys@hanyang.ac.kr

Received: 26 June 2023 Accepted: 26 October 2023 Published: 19 March 2024

ABSTRACT

Much like humans focus solely on object movement to understand actions, directing a deep learning model's attention to the core contexts within videos is crucial for improving video comprehension. In the recent study, Video Masked Auto-Encoder (VideoMAE) employs a pre-training approach with a high ratio of tube masking and reconstruction, effectively mitigating spatial bias due to temporal redundancy in full video frames. This steers the model's focus toward detailed temporal contexts. However, as the VideoMAE still relies on full video frames during the action recognition stage, it may exhibit a progressive shift in attention towards spatial contexts, deteriorating its ability to capture the main spatio-temporal contexts. To address this issue, we propose an attention-directing module named *Transformer Encoder Attention Module* (TEAM). This proposed module effectively directs the model's attention to the core characteristics within each video, inherently mitigating spatial bias. The TEAM first figures out the core features among the overall extracted features from each video. After that, it discerns the specific parts of the video where those features are located, encouraging the model to focus more on these informative parts. Consequently, during the action recognition stage, the proposed TEAM effectively shifts the VideoMAE's attention from spatial contexts towards the core spatio-temporal contexts. This attention-shift manner alleviates the spatial bias in the model and simultaneously enhances its ability to capture precise video contexts. We conduct extensive experiments to explore the optimal configuration that enables the TEAM to fulfill its intended design purpose and facilitates its seamless integration with the VideoMAE framework. The integrated model, i.e., VideoMAE + TEAM, outperforms the existing VideoMAE by a significant margin on Something-Something-V2 (71.3% *vs.* 70.3%). Moreover, the qualitative comparisons demonstrate that the TEAM encourages the model to disregard insignificant features and focus more on the essential video features, capturing more detailed spatio-temporal contexts within the video.

KEYWORDS

Video classification; action recognition; vision transformer; masked auto-encoder

1 Introduction

The transformer [1] has made significant progress in the Natural Language Processing (NLP) field because of its self-attention mechanism, which enables seamless modeling of long-term dependencies. In addition, the Vision Transformer (ViT) architecture [2] is widely used in computer vision tasks, such as image classification, object detection, semantic segmentation, and video understanding, showing superior performance compared to to its convolutional counterparts [3–7]. This performance demonstrates that self-attention can smoothly capture global dependencies, not only in natural language but also in images or videos–both spatially and temporally.

In the process of training a transformer, it is more effective to carry out pre-training on a large-scale dataset to improve generalization performance and then conduct fine-tuning on the target dataset, rather than directly train the transformer from scratch. In the image domain, it is common to pre-train large-scale image datasets, such as ImageNet-21k or JFT-300 M. However, in the video domain, existing datasets are relatively small compared to image datasets, making it challenging to pre-train directly on them. Therefore, a typical approach to effectively training video transformers is to pre-train on a large-scale image dataset and then fine-tune it on the video dataset. However, since the video models are pre-trained on large-scale image datasets, they tend to understand each video more based on its spatial context than its temporal context. In other words, the video transformer is more likely to be proficient in spatial modeling, but it may have limitations in temporal modeling. To address this issue, recent approaches [8–11] typically involve self-supervised methods on small-scale video datasets to pre-train and fine-tune the model on the target dataset, resulting in enhanced temporal modeling.

In Self-Supervised Video Pre-training (SSVP), the VideoMAE [8] significantly enhances the temporal modeling ability in the ViT by changing only the pre-training method from image-based supervised learning to masking and reconstruction. Specifically, it employs a high ratio (i.e., 90%) of tube masking, which is designed to avoid “spatial bias” caused by temporal redundancy in the video. Basically, a video contains both spatial and temporal contexts, where the temporal context refers to the pattern of the spatial context evolving over time. In the video-training pipeline, each video is sampled with T frames, and the model is trained to understand the video’s characteristics based on these frames. However, as shown in Fig. 1, the spatial context within video frames varies slowly over time, leading to significant redundancy of irrelevant spatial information within the frames. Due to this redundancy, within consecutive frames, only a small portion is practically associated with the motion (e.g., twisting motion in Fig. 1), while the majority tends to consist of redundant spatial information (e.g., background or face in Fig. 1). Therefore, the temporal redundancy in full video frames may lead to a progressive shift in the model’s attention towards only the redundant spatial contexts. This spatial bias issue deteriorates the model’s ability to comprehend the detailed characteristics of the video, leading to a significant performance drop in action recognition. From this perspective, employing a high ratio of tube masking is an effective approach to directly mitigate the spatial bias issue and shift the model’s attention to the core characteristics of the video. In addition, since the VideoMAE aims to reconstruct the contextual information of masked tubes by utilizing computed features from only the unmasked ones, it attempts to extract as many features as possible from unmasked segments. Through this effective pre-training, the VideoMAE becomes capable of extracting substantial features from the input video, leading to an improved ability to capture the main spatio-temporal contexts.

However, in the action recognition stage, the VideoMAE may not fully utilize the benefits of masking and reconstruction as it still relies on full video frames that inherently contain temporal redundancy. During the initial epochs of fine-tuning, this effective pre-training enables the VideoMAE to capture spatio-temporal contexts in video seamlessly. However, as fine-tuning progresses, the model may progressively become focused solely on the redundant spatial contexts due to the inherent temporal redundancy in full video frames. Therefore, it is still challenging to effectively steer the model’s attention towards the core characteristics of each video, even in the fine-tuning stage, and utilize the potential advantage of SSVP.

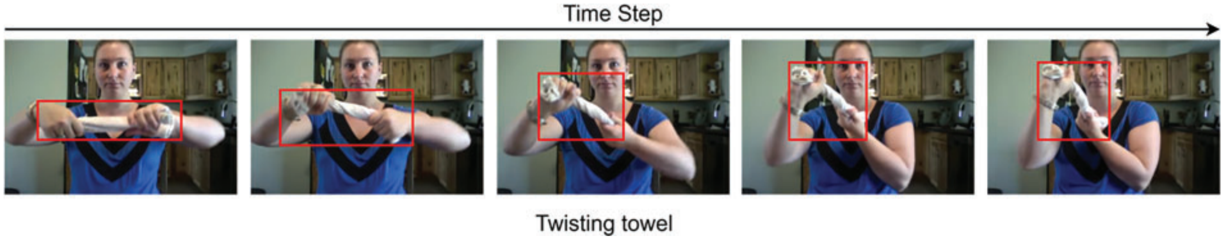

Figure 1: Consecutive frames from Something-Something-V2. Red boxes in these frames indicate motion-related area, i.e., twisting motion

To address this challenge, we propose a novel module named the *Transformer Encoder Attention Module* (TEAM). The proposed module aims to direct the attention of the VideoMAE to the core contexts of the video. In this way, it effectively mitigates the spatial bias issue by shifting the model's attention from spatial contexts to the main spatio-temporal contexts. Therefore, this shift in focus leads to a performance improvement in video understanding. To ensure the seamless integration of the TEAM with the VideoMAE, we first analyzed the video processing flow in the VideoMAE. Based on this analysis, we designed the TEAM, which comprises two sub-modules, each with a specific role in guiding the model's attention to the core contexts of each video, thereby mitigating the spatial bias issue. The first sub-module is the Feature Attention Module (FAM). It aims to identify core features among the overall features extracted by the VideoMAE, encouraging the model to focus more on those core features. Consequently, this sub-module prevents the model from being spatially biased in the action recognition stage. The second sub-module, i.e., the Token Attention Module (TAM), operates based on the FAM's output. It first discerns the meaningful parts of the video where the informative features are located and then enables the VideoMAE to concentrate on those regions. This allows the model to capture more detailed spatio-temporal contexts in the video. We sequentially organized the FAM and TAM into the TEAM. As a result, the TEAM guides the VideoMAE to recognize *what and where the core features are* in the video, resulting in improved performance in terms of video understanding.

We designed the TEAM by taking inspiration from the CBAM [12], one of the several feature selection methods [13–15]. This CBAM is an attention module developed for image-based convolutional networks. The TEAM shares similarities with the CBAM as an attention module, but it is specifically developed for a video-based transformer, i.e., VideoMAE. Therefore, we conducted extensive experiments to explore the optimal structure of the TEAM that best fits the VideoMAE. To be more specific, we experimented by refining the existing CBAM structure and comparing it to find the most desirable structure that enables the FAM and TAM to effectively achieve their respective roles in the VideoMAE. Through these experiments, we observed two important findings: (1) The ratio of core features to the overall extracted features progressively decreases as video data passes through each encoder within the VideoMAE; (2) Using only the softmax function without any learnable parameters, e.g., the linear layer, is more effective in discerning the informative parts within a video. Building upon these findings, we developed the TEAM and confirmed that this design choice enables the TEAM to operate more effectively within each encoder of the VideoMAE. Consequently, we compared the integrated model, i.e., VideoMAE + TEAM, with the existing VideoMAE on Kinetics-400 (K400) [16] and Something-Something-V2 (SSv2) [17]. The integrated model demonstrates higher performance compared to the baseline on both datasets, with particularly notable improvements observed on SSv2 (71.3% *vs.* 70.3%). Lastly, we conducted qualitative comparisons, which indicate that the TEAM allows the model to better focus on the informative features, enabling it to capture more detailed spatio-temporal contexts from the video. In summary, the main contributions of this paper are as follows:

- We propose an attention-directing module, i.e., TEAM. This module guides the VideoMAE to figure out *what and where the core features are* in the video and enables the model to focus on those features. Consequently, it effectively addresses the spatial bias issue in the VideoMAE.
- We conduct extensive ablation studies to explore the optimal structure that enables seamless integration of the TEAM with the VideoMAE. Throughout these studies, we obtain two major findings: (1) As video data passes through each encoder within the VideoMAE, the ratio of core features to the overall extracted features gradually decreases; (2) Using only the softmax function is more effective in identifying the meaningful parts within a video than employing learnable parameters, i.e., the linear layer.
- Through performance and qualitative comparisons, we demonstrate that the TEAM enables the VideoMAE to better understand each video, leading to a significant performance improvement on SSv2. Since getting remarkable performance on SSv2 requires detailed video context modeling, this improvement proves the effectiveness of the TEAM.

The rest of this paper is organized as follows: In Section 2, we present an overview of the related work. Then we deeply describe the TEAM in Section 3. Section 4 shares the experimental results and analysis. Finally, we conclude this paper and present future work in Section 5.

2 Related Work

2.1 Transformer-Based Models in Action Recognition

The transformer [1] architecture has played a significant role in the NLP field due to its capability to capture long-term dependencies. Consequently, it has garnered substantial attention in the field of computer vision, and there have been numerous attempts to apply the ViT [2] architecture to various vision tasks. Moreover, as the video data is inherently sequential, this ViT architecture has garnered notable interest, particularly in action recognition. Video Vision Transformer [3] proposes a temporal encoder, while TimeSformer [6] introduces temporal attention. Each incorporates their methods into the ViT architecture, resulting in significantly improved performance in action recognition compared to their convolutional counterparts. Building upon this, transformer architectures have become widely used in the field of action recognition. For instance, the Video Swin Transformer [18] employs a window shift attention mechanism to capture the locality and activate the inductive bias of the transformer. MViT [19] utilizes a hierarchical structure and a pooling attention to reduce the computational cost. Uniformer [20] combines the strengths of convolutional networks and transformer-based models. And the recent work, Multi View Transformer [21], leverages multiple ViT structures to capture spatio-temporal contexts in various ways.

However, since these transformer-based models have less inductive bias compared to their convolutional counterparts, a common general scheme is to pre-train them on large-scale datasets and then fine-tune them on target datasets. Furthermore, due to the relatively small scale of video datasets, the early action recognition scheme typically involves pre-training the transformer on a large-scale image dataset and fine-tuning it on a video dataset. However, this image-based pre-training may bias the model towards only the spatial contexts in the video, potentially limiting its ability to seamlessly capture spatio-temporal contexts. To address this issue, recently proposed approaches [8–11] typically employ self-supervised video pre-training, enabling video-based models to better understand each video itself.

2.2 Self-Supervised Video Pre-Training

In self-supervised video pre-training, the VideoMAE [8] is a method that applies SSVP to the action recognition scheme for the first time. Instead of image-based supervision, it employs masking and reconstruction for pre-training, resulting in the greatly enhanced temporal modeling ability of the ViT. Likewise, the MaskFeat [11] exploits the masking and reconstruction scheme during SSVP. This method differs from the VideoMAE in that it predicts the HOG map of each video frame rather than reconstructing RGB pixels, and its masking ratio is lower (i.e., 60%). BEVT [22] also utilizes SSVP with masking and reconstruction to improve the spatio-temporal modeling ability of the Video Swin Transformer. However, it differentiates itself from the above methods by predicting features from a trained tokenizer, similar to BERT's pre-training approach [23]. Most recently, the InternVideo [9] utilized SSVP with both masking and reconstruction, as well as contrastive learning, achieving state-of-the-art performance on most video under standings tasks.

Consequently, recent methods typically employ self-supervised learning during pre-training to enhance the video understanding capability of existing video-based models. These approaches effectively address the limitations of image-based pre-training. Therefore, they achieve significantly higher performance on action recognition compared to the elaborately designed image-based video models. However, despite the effectiveness of SSVP, the action recognition scheme still relies on full video frames, which inherently contain temporal redundancy. This inefficient scheme may progressively lead SSVP-based models to become focused only on redundant spatial contexts as the fine-tuning continues. To address this issue, we developed the TEAM, which redirects the model's attention to the main spatio-temporal contexts in each video. By doing so, the proposed module enables the SSVP-based model to better capture the core contexts in the video without becoming spatially biased, even as fine-tuning progresses. Unfortunately, among SSVP-based models, only the pre-trained weight of the VideoMAE on SSv2 is publicly available, and we have limited resources (i.e., GPUs). Therefore, we applied the TEAM to only the VideoMAE and evaluated its effectiveness.

2.3 Convolutional Block Attention Module

Classification is a task in which a model extracts features from data and then computes the main context based on these features, ultimately classifying it into a specific category. Moreover, prioritizing core features over unnecessary ones (e.g., background in an image) during feature extraction enhances the clarity of the computed main contexts, leading to improved classification performance.

Based on this concept, Woo et al. [12] proposed a *Convolutional Block Attention Module* (CBAM) which is an effective attention module for convolutional networks in image understanding tasks. Specifically, the authors designed CBAM with two sub-modules to precisely identify *what and where the core features are* within the feature map that represents the image. The first sub-module is the Channel Attention Module (CAM). It exploits inter-channel relationships to identify which channels in the feature map contain informative features, allowing the model to focus more on those core channels. The second sub-module, the Spatial Attention Module (SAM), operates based on the output of the CAM. It utilizes inter-spatial relationships to distinguish the spatial locations where core features are located. This makes the model focus more on the critical spatial regions within the entire spatial domain of the feature map. Through these sophisticated modules, CBAM significantly improves the context modeling ability of the existing convolutional networks, thereby enhancing their performance in image understanding tasks like object detection and image classification.

Inspired by CBAM, we propose the TEAM, which effectively enables the VideoMAE to focus more on the target features of the video. Since a video is sequential data in which the spatial contexts within each frame change over time, its core features should include not only spatial features but also temporal features. In other words, unlike the CBAM, which only considers the informative spatial features in one image, we should design the TEAM to make the model focus well on both the key spatial and temporal features in the video. Therefore, while the proposed TEAM consists of two sub-modules similar to the CBAM, we refined the components of each module to enable the TEAM to better identify the core features in each video.

3 Methodology

3.1 Video Representation in Transformer

The purpose of an attention module (e.g., CBAM or TEAM) is to guide the model to focus more on the informative features among the extracted data features, enabling it to capture more precise contexts in the data. Therefore, an integrated model with this type of module may better understand the data itself. To ensure that an attention module effectively operates for its intended purpose, it is essential to consider how the model processes data. By doing so, an attention module can be designed based on this analysis, ultimately enabling it to effectively complement the context modeling method of the overarching model. To achieve this objective, we first analyzed the video processing flow in the VideoMAE from a different perspective. Based on this analysis, we designed the TEAM to seamlessly integrate with the VideoMAE and effectively operate for its intended purpose.

The VideoMAE first receives each clip video as frames and converts them into a 3D token sequence. To be more specific, a video clip is first expressed as T frames of $\mathrm{H} \times W \times 3$ pixels through frame sampling. After that, the video, $\mathrm{V} \in \mathbb{R}^{\mathrm{T}\times\mathrm{H}\times\mathrm{W}\times 3}$, is decomposed into N non-overlapping 3D tokens through tube let embedding (i.e., $\mathrm{N} = \frac{\mathrm{T}}{\mathrm{t}} \times \frac{\mathrm{H}}{\mathrm{P}} \times \frac{\mathrm{W}}{\mathrm{P}}$), and each 3D token $x \in \mathbb{R}^{\mathrm{t}\times\mathrm{P}\times\mathrm{P}\times 3}$ is mapped into a D-dimensional vector. Lastly, a learnable positional embedding matrix, $\mathrm{E}_{pos} \in \mathbb{R}^{\mathrm{N}\times\mathrm{D}}$, is added to this 3D token sequence, encoding the spatio-temporal position of each individual 3D token. As a result, in the VideoMAE, each video is represented as a 3D token sequence $\mathrm{X} \in \mathbb{R}^{\mathrm{N}\times\mathrm{D}}$, and each 3D token is depicted as follows:

$$x_n = \left[x_{(n,\,1)},\ x_{(n,\,2)},\ x_{(n,\,3)}, \cdots\cdots\cdots,\ x_{(n,\,D)}\right] \tag{1}$$

As shown in Eq. (1), each 3D token consists of D values, and from the perspective of the latent space, x_n would be expressed as a single point in D-dimensional space. In this case, each dimension in the D-dimensional space represents a specific feature descriptor, and the value for the dimension is the degree to which a token contains that feature. Taking Eq. (1) as an example, if the first dimension represents the feature descriptor "going up", it can be interpreted that a 3D token x_n contains the "going up" feature to the degree of $x_{(n,1)}$.

Based on this concept, each clip is converted into a 3D token sequence $\mathrm{X} \in \mathbb{R}^{\mathrm{N}\times\mathrm{D}}$ by tubelet embedding, and it is expressed as N points in a D-dimensional space. This token sequence passes through a number (L) of encoders within the VideoMAE, and the input and output sequences of each encoder have the same shape of $\mathrm{N} \times \mathrm{D}$. From the perspective of the latent space, this token sequence would be represented as N points in different D-dimensional spaces as it passes through each encoder. At this point, since the early encoders extract relatively simple features from the video, the output sequence of these encoders would be projected into a latent space where each dimension represents those simple features. Furthermore, these simple features are combined through a weighted summation, allowing the later encoders to extract more sophisticated and complex features from the video. Therefore, as the encoder deepens, the output sequence would be projected into a latent space where each dimension represents more refined and intricate features. Finally, the output sequence of the final encoder within the VideoMAE is summarized into a D-dimensional vector through average pooling. Subsequently, a hidden layer, MLP Head, and Softmax are applied to this vector to classify it into a specific category. To summarize, we analyze the video processing flow in the VideoMAE from the perspective of the latent space. This model maps each video as N points in a latent space where each dimension describes a specific feature and processes it through L encoders. Based on this analysis, we design the TEAM in such a way that it allows the model to focus more on core features, enabling it to capture more precise video contexts.

3.2 Feature Attention Module

The Feature Attention Module (FAM) aims to figure out the core features of video, enabling the Video MAE to focus more on those features. Consequently, this sub-module effectively prevents the model from being spatially biased. The overall process of the FAM is depict in Fig. 2. We design it in three steps to ensure that the FAM operates appropriately for its intended purpose.

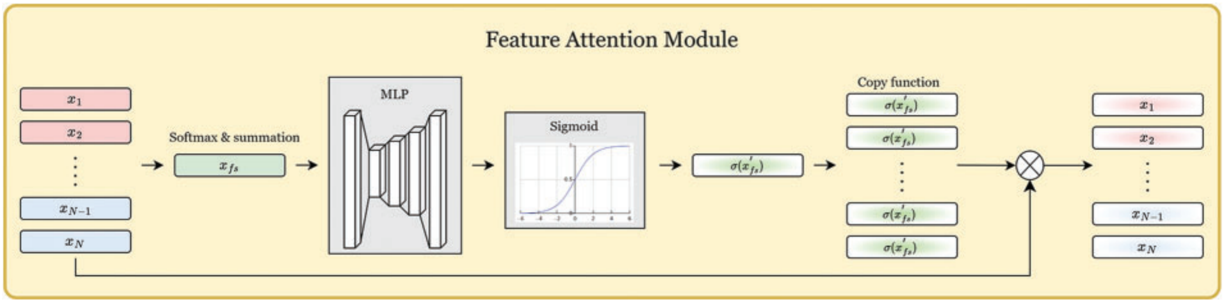

Figure 2: Overall process of the Feature Attention Module

3.2.1 Summarize the Token Sequence

To identify which features among the overall features of the video are informative for video understanding, it is necessary to first figure out what features the video contains. As mentioned in Section 3.1, the VideoMAE first converts a video into a 3D token sequence through tubelet embedding, and this sequence can be depicted as N points in D-dimensional space. In other words, since each token in the token sequence has a different value within the same feature descriptor (i.e., dimension), it is difficult to approximately recognize what degree this 3D token sequence has to each feature. Therefore, the FAM first summarizes the token sequence $\mathrm{X} \in \mathbb{R}^{\mathrm{N}\times\mathrm{D}}$ as follows:

$$x_{fs} = \mathrm{sum}_{\mathrm{FD}}\,(\mathrm{softmax}\,(\mathrm{X}) \otimes \mathrm{X}) \tag{2}$$

Here, $\otimes$ denotes element-wise multiplication, and $\mathrm{sum}_{\mathrm{FD}}$ indicates the summation function along the feature descriptor axis. The softmax function is first applied to each feature descriptor axis in the token sequence. This process ensures the sum of N different values within the same feature descriptor becomes one. After that, the softmax output is element-wise multiplied with the input token sequence and then summed along the feature descriptor axis. Through this process, the token sequence $\mathrm{X} \in \mathbb{R}^{\mathrm{N}\times\mathrm{D}}$ is summarized into a single summarized token $x_{fs} \in \mathbb{R}^{1\times\mathrm{D}}$. This summarized token roughly represents the extent to which the video includes each feature.

3.2.2 Identify the Informative Features in the Video

To identify the core features in the video, the FAM processes the summarized token x_{fs} as follows:

$$x_{fs}^{'} = MLP\left(x_{fs}\right) = \mathrm{W}_3\left(\mathrm{W}_2\left(\mathrm{W}_1\left(\mathrm{W}_0\left(x_{fs}\right)\right)\right)\right) \tag{3}$$

Here, *MLP* is the multi-layer perceptron, and the GeLU activation function is followed by W_0, W_1, and W_2. The MLP in this sub-module is designed with an encoding and decoding structure, effectively filtering out poor features that are not related to the main contexts in the video. During the encoding process, the D-dimensional vector x_{fs} is mapped into a single point in a lower dimensional space through the hidden layer $\mathrm{W}_0 \in \mathbb{R}^{\mathrm{D}\times \mathrm{r}_1\cdot \mathrm{D}}$. Here, r_1 is the reduction ratio, which takes on a value between zero and one. In this process, W_0 is likely to be trained to enable the x_{fs} to be well expressed, even in the lower-dimensional space, while minimizing its information loss. In other words, as training progresses, the W_0 is expected to be progressively updated to map the x_{fs} into a lower-dimensional space while preserving only informative features among the overall features the x_{fs} has. After that, the encoded vector is decoded back into a D-dimensional vector through three hidden layers, i.e., $\mathrm{W}_1 \in \mathbb{R}^{\mathrm{r}_1\cdot \mathrm{D}\times \mathrm{r}_2\cdot \mathrm{D}}$, $\mathrm{W}_2 \in \mathbb{R}^{\mathrm{r}_2\cdot \mathrm{D}\times \mathrm{r}_3\cdot \mathrm{D}}$, and $\mathrm{W}_3 \in \mathbb{R}^{\mathrm{r}_3\cdot \mathrm{D}\times \mathrm{D}}$, and the relationship among these reduction ratios is as follows: $\mathrm{r}_1 < \mathrm{r}_2 < \mathrm{r}_3$. Here, each of the r_1, r_2, and r_3 has a value between zero and one. At this point, it can be interpreted that the encoded vector, composed of only the informative features, is sequentially mapped back to the D-dimension. Consequently, the decoded vector $x_{fs}^{'}$ is predicted to have high values for dimensions describing meaningful features, while exhibiting relatively low values for dimensions representing irrelevant features.

3.2.3 Integrate the Attention Output with the Input Sequence

Finally, to incorporate the information the $x_{fs}^{'}$ carries into the input token sequence, the FAM processes it as follows:

$$\mathrm{X}' = \mathrm{f}\left(\sigma\left(x_{fs}^{'}\right)\right) \otimes \mathrm{X} \tag{4}$$

Here, σ, and $\mathrm{f}()$ denote the sigmoid and copy functions, respectively. The FAM applies the sigmoid function to $x_{fs}^{'}$ which causes the degrees corresponding to dimensions expressing core features to converge to one, while the values for other dimensions converge to zero. This adjustment creates an attention-reference vector. Afterwards, to apply the D-dimensional attention-reference vector $\sigma\left(x_{fs}^{'}\right) \in \mathbb{R}^{1\times D}$ to the input token sequence, the FAM copies this vector N times, thereby obtaining an attention-reference sequence $\mathrm{f}\left(\sigma\left(x_{fs}^{'}\right)\right) \in \mathbb{R}^{N\times D}$. Finally, this attention-reference sequence is element-wise multiplied with the input token sequence, resulting in the final output X' of the FAM.

Compared to the input sequence X, the degrees of the core feature descriptors in the X' are expected to be maintained, but the values corresponding to dimensions depicting poor features would be decreased. In other words, as the token sequence X passes through each encoder integrated with the FAM, the degrees of poor feature descriptors within X gradually decrease, eventually approaching values close to zero. On the contrary, the degrees of core feature descriptors remain preserved. This progressive adjustment effectively directs the model's attention towards only the core features of the video. To summarize, the FAM identifies the core features within the video and gradually vanishes the other poor features, guiding the model to focus on those core features. Therefore, this sub-module effectively mitigates the spatial bias issue in the Video MAE through this attention shift manner.

3.3 Token Attention Module

The purpose of the Token Attention Module (TAM) is to distinguish meaningful tokens within the token sequence, enabling the model to focus more on those tokens. This allows the VideoMAE to capture more accurate spatio-temporal contexts in each video. To be more specific, the TAM operates based on the FAM's output X′, and it recognizes which tokens contain the core features identified by the FAM. Accordingly, it also makes the model ignore the other tokens that do not possess those features. The overall process of TAM is shown in Fig. 3. Unlike the FAM, the TAM does not use any of the learnable parameters. Consequently, we designed the TAM in two steps to ensure its effective operation for its intended purpose.

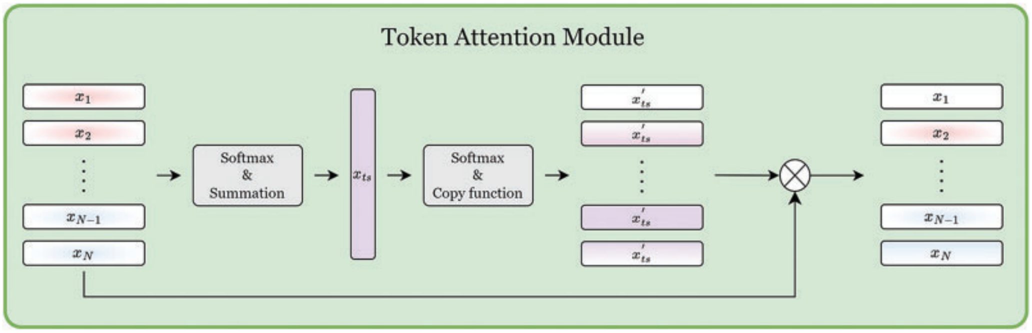

Figure 3: Overall process of the Token Attention Module

3.3.1 Discern Meaningful Tokens in the Token Sequence

The primary objective of the FAM is to recognize the key features among all the features of a video and encourage the model to concentrate on those features. Consequently, through this sub-module, the output sequence is expected to exhibit low values for poor feature descriptors while displaying relatively high values for informative feature descriptors. In addition, since each token contains different information, poor tokens are typically predicted to have low values across most feature descriptors, including those related to core features. Conversely, meaningful tokens are expected to have relatively higher values, specifically for informative feature descriptors. Therefore, we considered the magnitude of the D values constituting each token as the standard for distinguishing its meaningfulness. Accordingly, to efficiently identify meaningful tokens in the X′, the TAM first summarizes each token as follows:

$$x_{ts} = \mathrm{sum}_{\mathrm{T}}\left(\mathrm{softmax}\left(\mathrm{X}'\right) \otimes \mathrm{X}'\right) \tag{5}$$

Here, $\mathrm{sum}_{\mathrm{T}}$ denotes the summation function for each token. In particular, the TAM first applies the softmax function to each token in X′ and performs element-wise multiplication with the input X′. Afterward, it conducts summation along each token to summarize a token $x_n \in \mathbb{R}^{1\times \mathrm{D}}$ as an $x_{ts,\,n} \in \mathbb{R}^{1\times 1}$, ultimately converting the summary of $\mathrm{X}' \in \mathbb{R}^{\mathrm{N}\times \mathrm{D}}$ into the $x_{ts} \in \mathbb{R}^{\mathrm{N}\times 1}$. At this point, the value of each summarized token has no specific meaning. However, when comparing the values of each token with one another, a token with a relatively higher value compared to other tokens can be interpreted as an informative token. Therefore, the relative magnitude of each summarized token value is regarded as a key factor in determining meaningful tokens. Based on this concept, unlike the FAM, the TAM does not use any of the learnable parameters (i.e., weight and bias). Instead, it utilizes only the softmax function to discern the meaningfulness of each token:

$$x'_{ts} = \text{softmax}\left(x_{ts}\right) \tag{6}$$

Since the softmax function, unlike the sigmoid function, maps values using a relative standard, we considered it a suitable function for discerning whether each token is informative or not. As a result, in the softmax output $x'_{ts} \in \mathbb{R}^{\mathrm{N}\times 1}$, a core token is likely to have a relatively high value, while a poor token is expected to have a value closer to zero.

3.3.2 Integrate the Attention Output with the Input Sequence

Lastly, the TAM applies the x'_{ts} to the X′ to encourage the model to direct its attention to the meaningful tokens. It processes x'_{ts} as follows:

$$\mathrm{Y} = \mathrm{g}\left(x'_{ts}\right) \otimes \mathrm{X}' \tag{7}$$

Here, g() denotes the copy function. The TAM copies each summarized token D times to convert each token into a 1 × D shape, resulting in the $\mathrm{g}\left(x'_{ts}\right) \in \mathbb{R}^{N\times D}$, i.e., attention-reference sequence. It then performs element-wise multiplication between this result and the input sequence X′ to derive the final output Y. Compared to the input sequence X′, the poor tokens in the Y are typically expected to have lower values, while the values of the meaningful tokens are more likely to be maintained. To summarize, similar to the attention shift manner of the FAM, this sub-module first discerns the meaningful tokens in the X′ and gradually vanishes the other poor tokens. By doing so, the TAM enables the model to concentrate solely on informative tokens, facilitating detailed spatio-temporal modeling from the video.

3.4 Transformer Encoder Attention Module

We designed the TEAM by sequentially placing the FAM and then the TAM to ensure the effective operation of the TEAM for its intended objective. Therefore, the TEAM processes the input sequence X as follows:

$$\mathrm{Y} = \mathrm{TAM}\left(\mathrm{FAM}\left(\mathrm{X}\right)\right) \tag{8}$$

Through this arrangement, as shown in Fig. 4, the TEAM first utilizes the FAM to figure out which features from the input sequence X are associated with the main spatio-temporal context. Subsequently, it gradually vanishes the features unrelated to the main context, allowing the model to concentrate more on the identified features, i.e., core features. Following this, using the TAM, the TEAM distinguishes the meaningful tokens containing the core features identified by the FAM. It then progressively reduces the values of tokens that do not contain core features while maintaining the tokens containing those features, resulting in the final output Y. Consequently, the TEAM progressively guides the model to focus more on the meaningful areas related to the main spatiotemporal context within video frames. This gradual adjustment effectively steers the attention of the VideoMAE from redundant spatial contexts to the core contexts of the video, mitigating the spatial bias issue in the model. As a result, the proposed module, i.e., TEAM, enhances the model's video context modeling capabilities and ultimately leads to improved performance in video understanding.

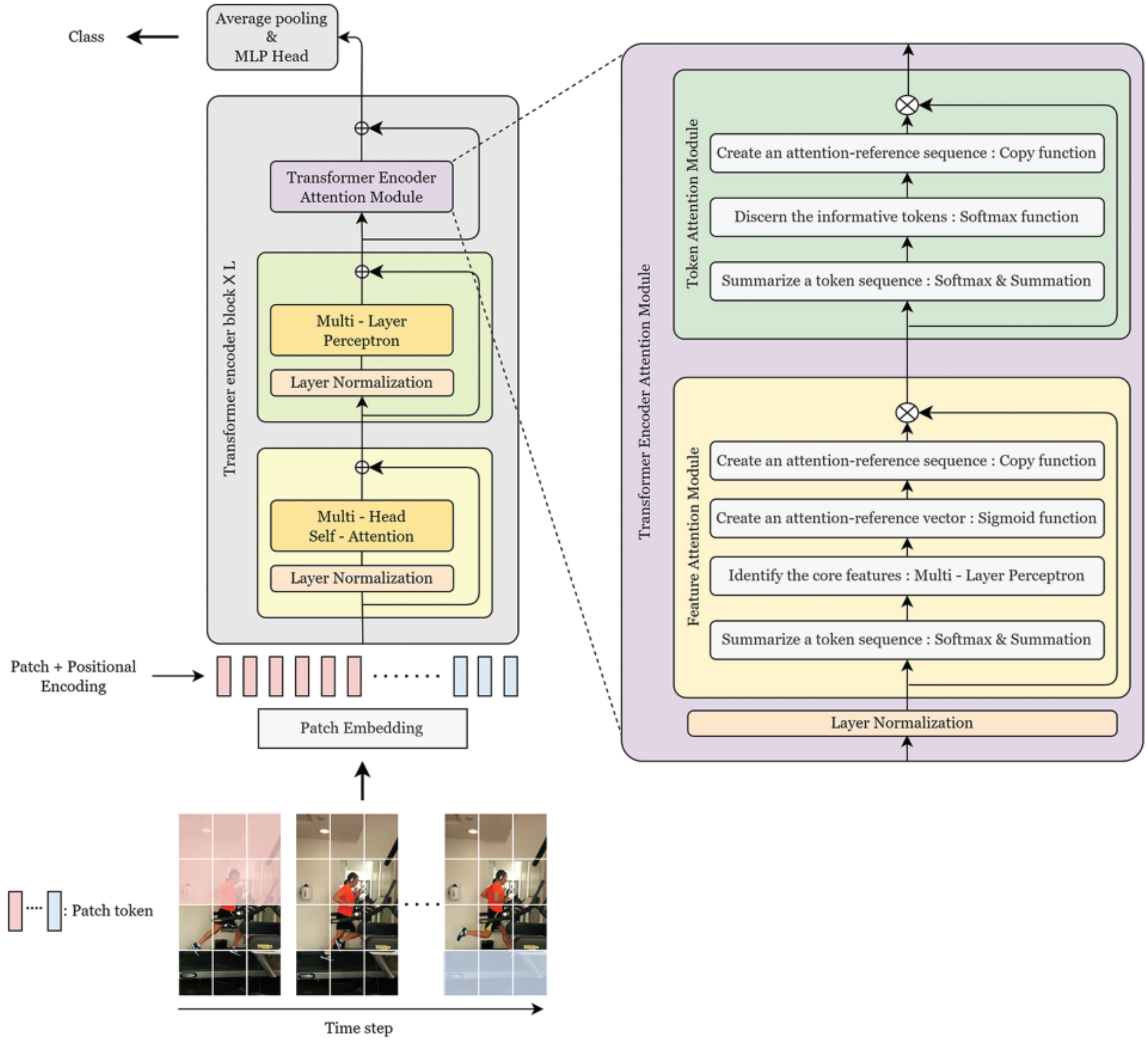
Class
Average pooling
&
MLP Head
Transformer Encoder
Attention Module
Multi - Layer
Perceptron
Layer Normalization
Multi - Head
Self - Attention
Layer Normalization
Transformer encoder block X L
Patch + Positional
Encoding
Patch Embedding
: Patch token
Time step
Transformer Encoder Attention Module
Token Attention Module
Create an attention-reference sequence : Copy function
Discern the informative tokens : Softmax function
Summarize a token sequence : Softmax & Summation
Feature Attention Module
Create an attention-reference sequence : Copy function
Create an attention-reference vector : Sigmoid function
Identify the core features : Multi - Layer Perceptron
Summarize a token sequence : Softmax & Summation
Layer Normalization

Figure 4: Overall architecture of the integrated model, i.e., VideoMAE + TEAM, and flow chart of the TEAM

Fig. 4 shows the overall architecture of the integrated model, i.e., VideoMAE + TEAM. We applied the TEAM at the back of each encoder, including a residual connection and a layer normalization layer. As a result, the TEAM recognizes core features from the overall features, extracted by self-attention and the multi-layer perceptron, and discerns informative tokens that include those features. The early encoders in the VideoMAE typically compute low-level features while progressively changing to extract high-level features as the encoder deepens. Based on this concept, we conducted various experiments. We observed that, as the position of the encoder deepens, the ratio of main context-related features decreases in the overall features extracted from the input sequence. Therefore, when applying the TEAM to each encoder at a different depth, we adjusted the hyper-parameters of the MLP in the FAM differently. Moreover, we applied the TEAM to only the last eight encoders. The reasons for and analysis of this selective application are discussed in Section 4.

4 Experiment

4.1 Experimental Setup

4.1.1 Implementation Details

Due to the limited available resources (i.e., GPU) in our lab and the fact that only the pre-trained checkpoints of the VideoMAE-Base and Small are publicly available, we integrated the TEAM with these models and evaluated the effectiveness of our module. Furthermore, through extensive experiments, we observed that the TEAM becomes more effective when integrated with the later encoders in the VideoMAE. Therefore, we evaluated its performance by applying it to only the last eight encoders (instead of all 12 encoders). Moreover, as the depth of the encoder deepens, the extracted features become more sophisticated. This feature-level change may lead to an alteration in the ratio of the core features relative to the overall extracted features. Hence, we adjusted the hyper-parameters (i.e., r_1, r_2, and r_3) differently in the MLP of the FAM according to the depth of each encoder. Specifically, we set up r_1, r_2, and r_3 differently for every four encoders, and these settings are shown in Table 1.

Table 1: We adjusted the settings of $\mathbf{r}_1$, $\mathbf{r}_2$, and $\mathbf{r}_3$ differently for every four encoders. This table shows the setting of the MLP within the FAM corresponding to each depth. The meaning of these hyper-parameters is discussed in Section 4.4

Model	Dimension (D)	Depth	Reduction ratio (r)	
VideoMAE-Small	384	5~8	r_1	0.2
			r_2	0.4
			r_3	0.6
		9~12	r_1	0.08
			r_2	0.16
			r_3	0.32
VideoMAE-Base	768	5~8	r_1	0.225
			r_2	0.45
			r_3	0.675
		9~12	r_1	0.1
			r_2	0.2
			r_3	0.4

If the calculation result of $r_1 \cdot D$ does not yield an integer, we round it to the nearest integer before applying it. For instance, if the r_1 is 0.2 in the VideoMAE-Small, we set the $r_1 \cdot D$ to be 77. As a result, based on the settings shown in Table 1, we designed the TEAM and applied it to the VideoMAE for evaluation. Through extensive experiments, we found that this setting allows the TEAM to operate most effectively for its intended purpose. More details on this are discussed in Section 4.4.

4.1.2 Dataset

We evaluated the effectiveness of the TEAM using the Kinetics-400 and Something-Something-V2 datasets, which contain different target contexts.

Kinetics-400 (K400) [16] is a large-scale video dataset consisting of 400 human action categories with a minimum of 400 clips per category. These videos are sourced from different YouTube videos and have an average duration of 10 s. Furthermore, most of these videos generally show human-object or human-human interactions and sports-related content. To identify the general target context within K400, we used Grad-CAM [24] to confirm the specific areas on which the trained VideoMAE focuses during video classification. Fig. 5a shows the target context the model concentrates on when classifying a "running on treadmill" video in K400. This qualitative result reveals that the VideoMAE pays more attention to the "treadmill" than the "running person". In other words, the model classifies the video based on spatial context rather than temporal context. Likewise, the common context across categories in K400 predominantly involves specific spatial information. As a result, the target context in K400 is well-known to be the spatial context.

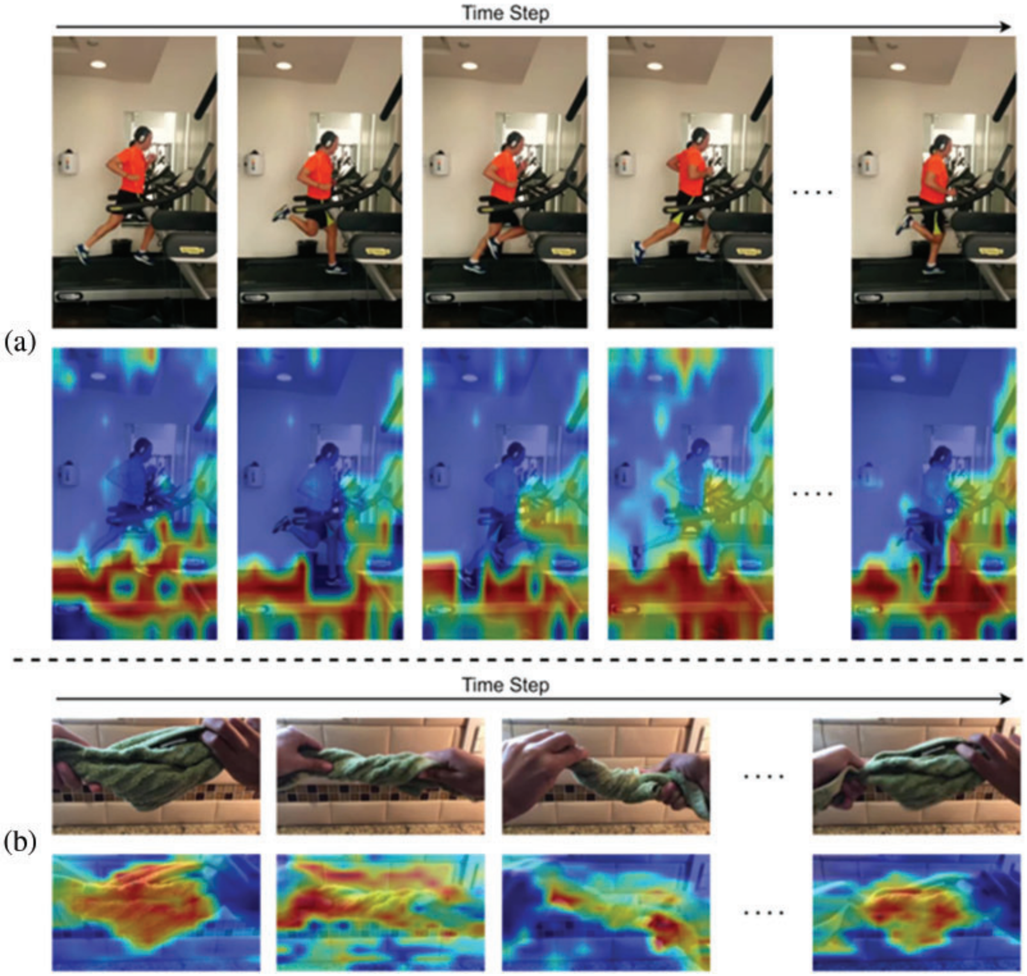

Figure 5: We utilized Grad-CAM [24] to visualize the areas that the trained VideoMAE focuses on when it classifies each video into a specific category. The resulting heat map indicates that the VideoMAE relies on the red regions when making classification decisions for each video frame

Something-Something-V2 (SSv2) [17] is a widely used video dataset comprising 220k videos across 174 categories. Each video in this dataset typically has a duration range from 2 to 6 s and expresses basic human action, such as "moving something up". Fig. 5b demonstrates the areas of focus for the trained VideoMAE when classifying a "twisting something" video in SSv2. The heat map shows that the model concentrates on the object directly associated with the twisting motion. This result indicates that the model focuses more on the spatio-temporal context of the video. Besides, SSv2 contains categories such as "dropping something into something" and "dropping something next to something". To classify these two categories well, the model should be able to capture detailed spatio-temporal contexts from each video. As a result, unlike K400, the target context in SSv2 is well-known to be detailed spatio-temporal information.

4.1.3 Training Details

To ensure a fair comparison, we integrated the TEAM with the VideoMAE and trained it using the same training scenario the existing VideoMAE used. However, due to our limited available resources (4 NVIDIA RTX A5000 GPUs) compared to the GPUs used by Tong et al. [8], we slightly adjusted the mini-batch size and learning rate from the original scenario. In VideoMAE-Small + TEAM, we set the learning rate and mini-batch size to 2.5×10^{-4} and 24, respectively. For VideoMAE-Base + TEAM, the learning rate is 1.768×10^{-4} and the mini-batch size is 12. In other words, apart from the learning rate and mini-batch size, we kept all other training parameters the same as those used for the original VideoMAE. Before the training, we applied the randomly initialized TEAM to the VideoMAE and trained it for its evaluation. More details are shown in Appendix A.

4.2 Main Results and Analysis

For each video, we sampled 16 frames composed of 224 × 224 × 3 pixels and used them for training. The tubelet size was 2 × 16 × 16 (i.e., t = 2 and P = 16), resulting in an input sequence shape of 1568×768 for VideoMAE-Base + TEAM and 1568×384 for VideoMAE-Small + TEAM. During inference, we first sampled 5 and 2 temporal clips from each K400 and SSv2 video, respectively. For each clip, we extracted 16 frames and applied three random spatial crops per frame. Consequently, we obtained the final prediction using 16 × 5 × 3 frames from a K400 video and 16 × 2 × 3 frames from an SSv2 video. The performance of the proposed TEAM on each dataset is shown in Tables 2 and 3.

Table 2: Performance comparisons with the vision transformer-based SSVP methods on Kinetics-400. All methods in this table are pre-trained on Kinetics-400, and each method processes a video with a shape of 16 × 224 × 224 × 3 in video classification. $\boldsymbol{\rho}$ is a masking ratio for VideoMAE and MAR

Method	Backbone	Param (M)	Top–1 (%)	Top–5 (%)
VideoMAE$_{\rho=90\%}$ [8]	ViT–S	22	79.0	93.8
VideoMAE$_{\rho=90\%}$ [8]	ViT–B	87	80.0	94.4
MAR$_{\rho=75\%}$ [25]	ViT–B	94	79.4	93.7
MAR$_{\rho=90\%}$ [25]	ViT–B	94	81.0	94.4
VideoMAE + TEAM (ours)	ViT–S	**23**	**79.0**	**93.9**
VideoMAE + TEAM (ours)	ViT–B	**91**	**80.2**	**94.4**

Table 3: Performance comparisons with the vision transformer-based SSVP methods on Something-Something-V2. All methods listed in this table are pre-trained on SSv2 and utilize a video shape of 16 × 224 × 224 × 3 for video classification. The parameter $\boldsymbol{\rho}$ denotes the masking ratio applied in the VideoMAE and MAR

Method	Backbone	Param (M)	Top–1 (%)	Top–5 (%)
VideoMAE$_{\rho=90\%}$ [8]	ViT–S	22	66.8	90.4
VideoMAE$_{\rho=90\%}$ [8]	ViT–B	87	70.3	92.4
MAR$_{\rho=75\%}$ [25]	ViT–B	94	69.5	91.9
MAR$_{\rho=90\%}$ [25]	ViT–B	94	71.0	92.8
VideoMAE + TEAM (Ours)	ViT–S	**23**	**67.6**	**90.5**
VideoMAE + TEAM (Ours)	ViT–B	**91**	**71.3**	**92.8**

Table 2 indicates how much the TEAM improves the video understanding performance of the Video-MAE on K400. It reveals that the integrated model with the TEAM shows a slight performance improvement over the VideoMAE. Fundamentally, the main objective of the TEAM is to prevent the VideoMAE from being spatially biased during fine-tuning and enable the model to capture more accurate video contexts. As discussed earlier, the spatial context of each category in K400 is easily distinguishable. This allows the model to achieve relatively high performance on this dataset by modeling only the spatial contexts. In other words, most videos in K400 primarily contain spatial information as their main context. Moreover, due to the temporal redundancy in full video frames, the focus of the VideoMAE is expected to become spatially biased as fine-tuning progresses. Therefore, when applying the TEAM, which operates in an attention-shift manner, to the VideoMAE and training it on the K400, the model's attention remains primarily on the spatial context rather than shifting to other important spatio-temporal contexts. This implies that the TEAM is not effectively operated on this dataset, leading to limited performance improvement in the VideoMAE (80.2% *vs.* 80.0%). Unlike K400, Table 3 reveals that the TEAM makes the VideoMAE improve its performance significantly on SSv2 (71.3% *vs.* 70.3%). This improvement suggests that the TEAM effectively mitigates spatial bias in the VideoMAE, allowing the model to capture detailed spatio-temporal context, which is the target context in SSv2. Therefore, we report that the TEAM enables the VideoMAE to effectively identify target contexts from each video and improve its performance on SSv2. To determine the optimal structure that allows the TEAM to effectively operate for its purpose, we conducted all ablation studies with the SSv2.

4.3 Performance Comparison with MAR

Similar to the design purpose of the TEAM, MAR [25] is an approach proposed to address the issue of spatial bias in the VideoMAE resulting from its inefficient action recognition scheme. It is not a specific model or module, but rather a training approach that improves the performance of the VideoMAE through efficient fine-tuning. This approach makes the VideoMAE maintain its masked auto-encoder structure, even in fine-tuning, and incorporates a "Bridging classifier" for action prediction. By doing so, the VideoMAE with MAR handles masked tokens by reconstructing them into RGB pixels, even in action recognition. Additionally, it classifies each video into a specific category based on computed contexts from unmasked tokens. Consequently, MAR employs a masking strategy to directly mitigate the spatial bias in the VideoMAE and effectively improves model performance through efficient fine-tuning.

Since the TEAM and MAR are proposed with the same intended purpose, we compared their performance to confirm which approach is more effective for the VideoMAE. In Tables 2 and 3, the TEAM shows worse performance on K400 than MAR in some cases but surpasses it on SSv2 (71.3% *vs.* 71.0%). MAR employs a masking strategy during action recognition to directly reduce the intrinsic characteristics of full video frames, including temporal redundancy. This direct approach effectively prevents the model from being spatially biased and yields superior performance on various video datasets. In contrast, the TEAM not only mitigates the spatial bias in the VideoMAE but also encourages the model to focus more on the core features of video, leading to improved context modeling. Therefore, in a spatially biased K400, the TEAM has lower performance compared to the MAR. However, in SSv2, which requires detailed spatio-temporal modeling, the TEAM shows superior results. Consequently, while MAR may be effective for directly reducing spatial bias in the model, the TEAM is expected to be more powerful in enhancing the context modeling capability of the model and improving overall performance on video understanding.

4.4 Ablation Study

The primary objective of the TEAM is to redirect the VideoMAE's attention from redundant spatial contexts to the main contexts of the video, thereby mitigating the spatial bias issue. In this case, we determined that the SSv2 dataset, with its main context being detailed spatio-temporal information, is suitable for conducting experiments to explore the optimal structure of the TEAM. Furthermore, as mentioned in Section 2.3, we drew inspiration from the functioning principles of the CBAM and designed the TEAM. To be more specific, we systematically investigated various design choices by modifying or altering the individual components of the CBAM, aiming to explore an attention-directing module best suitable for a video-based transformer, i.e., VideoMAE. Therefore, all ablation studies in this section provide empirical evidence supporting the effectiveness of our design choice. Moreover, all experiments are conducted based on SSv2, and we used the VideoMAE-Base as the baseline.

4.4.1 Analysis of Different Summary Methods

To identify the core features and meaningful parts in each video, the FAM and TAM in the TEAM first summarize the input sequence into x_{fs} and x_{ts}, respectively. In this case, it is crucial to summarize the input sequence for each purpose, as it enables each module to operate effectively. For instance, if the FAM poorly summarizes the input sequence into x_{fs}, it would be difficult for this module to identify the overall features the sequence has and, consequently, figure out the core features. Furthermore, since each video contains both spatial and temporal information, it is essential to minimize information loss when summarizing the token sequence representing a video. Therefore, we explored an optimal method that summarizes the input sequence while minimizing information loss.

Average pooling applies equal weights to all the values when summarizing, resulting in an equal emphasis on both salient and faint, less relevant parts. Therefore, using only this average pooling technique to summarize the input sequence may lead to dilution of the salient parts and significant information loss. To address this issue, CBAM [12] incorporates average pooling with max pooling to preserve the information from the most salient parts. However, this method also leads to information loss for somewhat salient parts, only emphasizing the most prominent information. To minimize the loss and summarize the input sequence effectively, we employed the softmax function to assign relatively high weights to salient parts and low weights to faint parts. We compared our summary method with the method used in CBAM to validate that the softmax & sum effectively summarizes the input sequence better than the combined pooling method. These results are presented in Table 4, indicating that the softmax & sum method shows a more proficient ability to summarize the token sequence compared to the method used in CBAM (71.3% *vs.* 71.1%). In other words, the softmax & sum method aims to summarize the input sequence while preserving as much information as possible, enabling the FAM and TAM to effectively perform for their respective purposes.

Table 4: Comparison of different summary methods. We observed that ours is more effective in summarizing each feature or token compared to the average pooling (AvgPool) + max pooling (MaxPool) method used in CBAM. We conducted these experiments with SSv2

Summary method		Top–1 (%)	Top–5 (%)
Feature Attention Module	Token Attention Module		
AvgPool + MaxPool	AvgPool + MaxPool	71.1	92.6
Softmax & sum (ours)	AvgPool + MaxPool	71.2	92.7
AvgPool + MaxPool	Softmax & sum (ours)	71.1	92.7
Softmax & sum (ours)	Softmax & sum (ours)	**71.3**	**92.8**

4.4.2 Analysis of Different Reduction Ratios

After the summarization, the FAM forwards the x_{fs} to an MLP with an encoding and decoding structure to identify core features in the video. During this process, the MLP first encodes the D-dimensional vector, x_{fs}, as a single point in the $\mathrm{r}_1 \cdot \mathrm{D}$-dimensional space. As training progresses, the encoding layer of this MLP is expected to be progressively trained to effectively project x_{fs} into a lower-dimensional space, $\mathrm{r}_1 \cdot \mathrm{D}$. In other words, this encoded vector is more likely to primarily consist of only the core features among the overall features of x_{fs}. At this point, as the reduction ratio, r_1, determines the dimension size to which the x_{fs} is mapped, we can consider the r_1 as a key point in deciding the proportion of these core features within the x_{fs}. Therefore, we conducted experiments by varying the reduction ratio, i.e., r_1, to investigate the proportion of meaningful features among all the extracted features in the VideoMAE. In this experiment, we configured the MLP with a single hidden layer (the MLP is composed of only two parameters, i.e., W_0 for encoding and W_1 for decoding) and applied the TEAM to the last four encoders within the VideoMAE. This design choice allows us to evaluate only the impact of a different reduction ratio.

According to the VideoMAE, the video inherently contains temporal redundancy, indicating that a significant proportion of the features within full video frames are irrelevant spatial information. Therefore, we assumed that the proportion of core features within the overall features would be less than 20%. Based on this concept, we set the reduction ratio, r_1, between 0.08 and 0.2. We expect the optimal value will lead to the highest performance. These results are shown in Table 5, demonstrating that approximately 10% of the overall features in a video are core features. If the reduction ratio is lower than 0.1, some of the core features might be omitted, resulting in information loss and a negative impact on performance. In contrast, if the reduction ratio exceeds 0.1, the FAM is expected to contain unimportant features (i.e., noise) among the core features it has determined, leading to a negative effect on performance.

Table 5: Comparison of different reduction ratios

Reduction ratio (r_1)	Top–1 (%)	Top–5 (%)
0.2	70.78	92.58
0.15	70.75	92.58
0.125	70.91	92.59
0.1	**70.93**	92.62
0.08	70.86	**92.64**

4.4.3 Analysis of the Relationship between Computed Feature Level and Proportion of Main Features

In general, the low-level encoders in a model extract relatively simple features from the data. Moreover, as the depth of layers increases, the extracted simple features undergo weight summation, making the high-level encoders compute more sophisticated and complex features. Following this concept, the VideoMAE focuses on extracting relatively simple features at the early encoders, while progressively changing to focus on computing complex features at the later encoders. Accordingly, we conducted extensive experiments to figure out the relationship between the variation of extracted feature levels with changing encoder depth and the proportion of core features. In particular, we adjusted the reduction ratio, r_1, of the TEAM differently as the encoder depth increased and observed the corresponding performance differences. In these experiments, we configured the MLP as a single hidden layer for a clear performance comparison.

Through the previous experiments, we empirically confirmed that the proportion of core features among the sophisticated features extracted by the last four encoders is 10%. Building upon this finding, we set the reduction ratio, r_1, of the TEAM to 0.1. We first applied this to the last four encoders within the VideoMAE. Subsequently, we extended the application range of the TEAM by applying it to two additional encoders, starting with the last four encoders. We continued this process until the TEAM was applied to all encoders and then measured the performance differences. These results are shown in Table 6, indicating that the performance is getting worse proportionally as we apply the TEAM to a greater number of encoders.

Interestingly, even when we apply the TEAM to all encoders, the performance turns out to be lower than the baseline, i.e., VideoMAE (70.3% *vs.* 69.5). We believe the main reason for this performance drop is that, in the early encoders, most of the extracted simple features are practically related to the main contexts in the video. However, even in these early encoders, we applied the TEAM with a reduction ratio of 0.1, resulting in significant information loss and consequently a negative impact on performance.

Table 6: Performance variation based on the depth of encoders with the TEAM applied. The symbol '×' denotes that the TEAM is not applied to the encoders at those depths

Depth						Top–1 (%)	Top–5 (%)
1∼2	3∼4	5∼6	7∼8	9∼10	11∼12		
0.1	0.1	0.1	0.1	0.1	0.1	69.5	91.9
×	×	0.1	0.1	0.1	0.1	70.5	92.5

(Continued)

Table 6 (continued)

Depth						Top–1 (%)	Top–5 (%)
1∼2	3∼4	5∼6	7∼8	9∼10	11∼12		
×	×	×	0.1	0.1	0.1	70.8	92.5
×	×	×	×	0.1	0.1	**70.9**	**92.6**
×	×	0.25	0.25	0.1	0.1	71.1	**92.8**
×	×	0.225	0.225	0.1	0.1	**71.2**	**92.8**
×	×	0.2	0.2	0.1	0.1	71.0	92.7
0.75	0.75	0.225	0.225	0.1	0.1	71.1	92.7
0.5	0.5	0.225	0.225	0.1	0.1	70.9	92.7

After that, to identify the approximate proportion of core features for each encoder depth, we adjusted the reduction ratio of the TEAM differently for each of the four encoders. Specifically, we fixed the reduction ratio at 0.1 for the last four encoders and varied it for the middle encoders, particularly the 5th to 8th encoders. In this case, we set the reduction ratio between 0.2 and 0.25. As shown in Table 6, we empirically observed that the proportion of core features among the overall mid-level features extracted by these encoders is 22.5%. These studies provide the empirical evidence that as the extracted features become simpler, the proportion of core features to the overall extracted features gradually increases. Lastly, we fixed the reduction ratio of the TEAM to 0.1 and 0.225 for the last and middle four encoders, respectively, and adjusted it for the initial four encoders. As shown in Table 6, using the TEAM for the initial four encoders has a negative impact on the performance. We believe the main reason for this negative impact is that the initial encoders tend to extract relatively simple features, all of which are indirectly associated with the main contexts in each video. Therefore, applying the TEAM to these encoders probably leads to information loss. As a result, these experiment results provide empirical evidence that as the depth of the encoder increases, the ratio of core features to the overall extracted features gradually decreases. Additionally, based on these experimental findings, we choose to apply the TEAM specifically to the last eight encoders within the VideoMAE.

4.4.4 Analysis of Different Decoding Configurations

We explored the optimal decoding configuration that effectively projects the encoded vector, x_{fs}, to a D-dimensional space while preserving its information. Specifically, we gradually incorporated a decoding layer into the MLP, comprising a single hidden layer, and observed the resulting performance difference. In Table 7, when the depth is 3 and $\lambda_{5\sim8}$ is [2,3], the MLP in the TEAM applied to the 5th to 8th encoders consists of three hidden layers. This allows the MLP to sequentially decode the encoded vector from an $r_1 \cdot D$-dimensional vector to an $r_2 \cdot D$ and an $r_3 \cdot D$, and consequently decode it to a D-dimensional vector. Here, r_2 and r_3 denote $2 \cdot r_1$ and $3 \cdot r_1$, respectively. Also, the GeLU activation function is followed by every hidden layer. These results are presented in Table 7. Based on these findings, we report that using multiple layers during the decoding process is effective in preserving the information of the encoded vector and seamlessly projecting it to a D-dimensional vector without any noise. Conversely, using too many layers for the decoding process may lead to overfitting and adversely affect generalization performance. Based on these experiments, we designed the MLP with three hidden layers for the TEAM.

Table 7: Comparison of different decoding configurations in the FAM. In this table, "Depth" refers to the number of hidden layers in the MLP, and $\boldsymbol{\lambda}_{a\sim b}$ denotes the decoding parameters used from a-th to the b-th encoder within the VideoMAE. Moreover, the symbol '×' indicates that the MLP used in this experiment consists of only one single hidden layer

Decoding configuration			Top–1 (%)	Top–5 (%)
Depth	$\lambda_{5\sim8}$	$\lambda_{9\sim12}$		
1	×	×	71.2	92.8
2	3	4	71.2	92.8
3	2, 3	2, 4	**71.3**	**92.8**
4	2, 3, 4	2, 4, 6	71.0	92.6

4.4.5 Analysis of Different Token Attention Methods

To identify informative features and meaningful spatial parts from the images, the CBAM employs a shared network and the sigmoid function for the Channel Attention Module (CAM) and the Spatial Attention Module (SAM). In this case, the shared network of the CAM and SAM is a single hidden layer MLP and a 7 × 7 convolution, respectively. These architectures guide the convolutional-based models to effectively focus on *what core features are* and *where they exist* in the image. However, we believe that such structures are not as effective in transformer-based models, such as the VideoMAE.

As mentioned in Section 3.3, the individual value of each summarized token does not carry any significant meaning. Instead, the relative magnitude of each token value determines its meaningfulness. There fore, unlike the CBAM, we did not use any learnable parameters to identify the meaningful parts in videos. Discerning the meaningfulness of each token necessitates comparing the relative sizes of the summarized token values. Based on this concept, we believe that the sigmoid function, which maps the values based on an absolute criterion, is not suitable for this module. On the contrary, since the softmax function maps each value based on a relative criterion, we considered it a suitable function for discerning the meaningfulness of each token. Therefore, we employed this softmax function in the TAM, enabling it to effectively distinguish meaningful tokens. Table 8 shows a performance comparison between our method and the conventional CBAM method when applied to the TEAM. It demonstrates that the identification of meaningful to kens can be effectively achieved using only the softmax function, without relying on any learnable para-meters. From these results, we conclude that using the softmax function alone allows transformer-based models to discern meaningful parts from videos more effectively.

4.4.6 Analysis of the Different Positions of the TEAM within Each Encoder

Finally, we applied the TEAM to the various positions in each encoder and compared the performance to explore the optimal application position. Since each encoder in the existing VideoMAE consists of self-attention and MLP components, we compared the performance of the TEAM applied at three different positions, as shown in Table 9. These results indicate that placing the TEAM at the last part of each encoder yields the best performance. The main objective of the TEAM is to identify the core features among the overall features extracted by the VideoMAE and distinguish meaningful tokens that contain these features. This enables the model to capture more accurate contexts from videos. Accordingly, we believe that integrating the TEAM after the feature extraction modules, i.e., self-attention and MLP, is more effective in achieving the intended purpose compared to other application cases. Therefore, we placed the TEAM after the self-attention and MLP modules in the last eight encoders of the VideoMAE.

Table 8: Comparison of different token attention methods in the TEAM. The MLP in this experiment consists of a single hidden layer for figuring out the meaningful parts of the video. Furthermore, the symbol '×' denotes that the TEAM consists of only one sub-module, which is the Feature Attention Module. The performance differences between the first method, i.e., excluding the TAM from the TEAM, and the second and third methods provide clear evidence of the TAM's significance as a sub-module for ensuring the effective operation of the TEAM

Method	Param (M)	Top–1 (%)	Top–5 (%)
×	91	70.9	92.5
MLP + Sigmoid	95	71.2	92.7
Softmax	**91**	**71.3**	**92.8**

Table 9: Performance comparison of different positions of the TEAM within each encoder

Encoder configuration	Top–1 (%)	Top–5 (%)
TEAM + Self-attention + MLP	71.2	92.7
Self-attention + TEAM + MLP	71.0	92.7
Self-attention + MLP + TEAM	**71.3**	**92.8**

4.5 Qualitative Comparison

To validate whether the TEAM operates effectively for its intended design purpose, we conducted visualizations for the VideoMAE and the integrated model, i.e., VideoMAE + TEAM. Specifically, we employed the Grad-CAM to observe the specific regions in SSv2 video where the models are focusing during the classification and compared their results. The model's focus on motion-related regions indicates that it concentrates mainly on the informative features in the videos. Therefore, the visualization results using the Grad-CAM can serve as valuable evidence to verify the effectiveness of the TEAM. Furthermore, we selected videos from different categories within the same action group for visualization to assess whether the TEAM enables the VideoMAE to capture detailed spatio-temporal contexts. For instance, in Fig. 6, all videos used for visualization are related to pulling motion but belong to different categories. To accurately classify these videos, the model should capture the detailed spatio-temporal contexts from each one. Therefore, we conducted visualizations using the videos from different categories within the same action group and calculated the precision score of each model. The results are shown in Figs. 6 and 7, demonstrating that the VideoMAE partially focuses on motion-related regions. These results also exhibit attention to irrelevant and meaningless regions. This poor attention may lead to the inclusion of irrelevant information during the modeling process of the main video contexts., thereby negatively affecting the performance of the model. In contrast, the integrated model, i.e., VideoMAE + TEAM, predominantly focuses on the motion-related regions while ignoring the irrelevant areas. This precise attention enables the integrated model to classify each video with a higher precision score compared to the existing VideoMAE. As a result, these visualization results demonstrate that the TEAM effectively guides the VideoMAE to concentrate on more the core features, enabling the model to accurately capture the main video contexts.

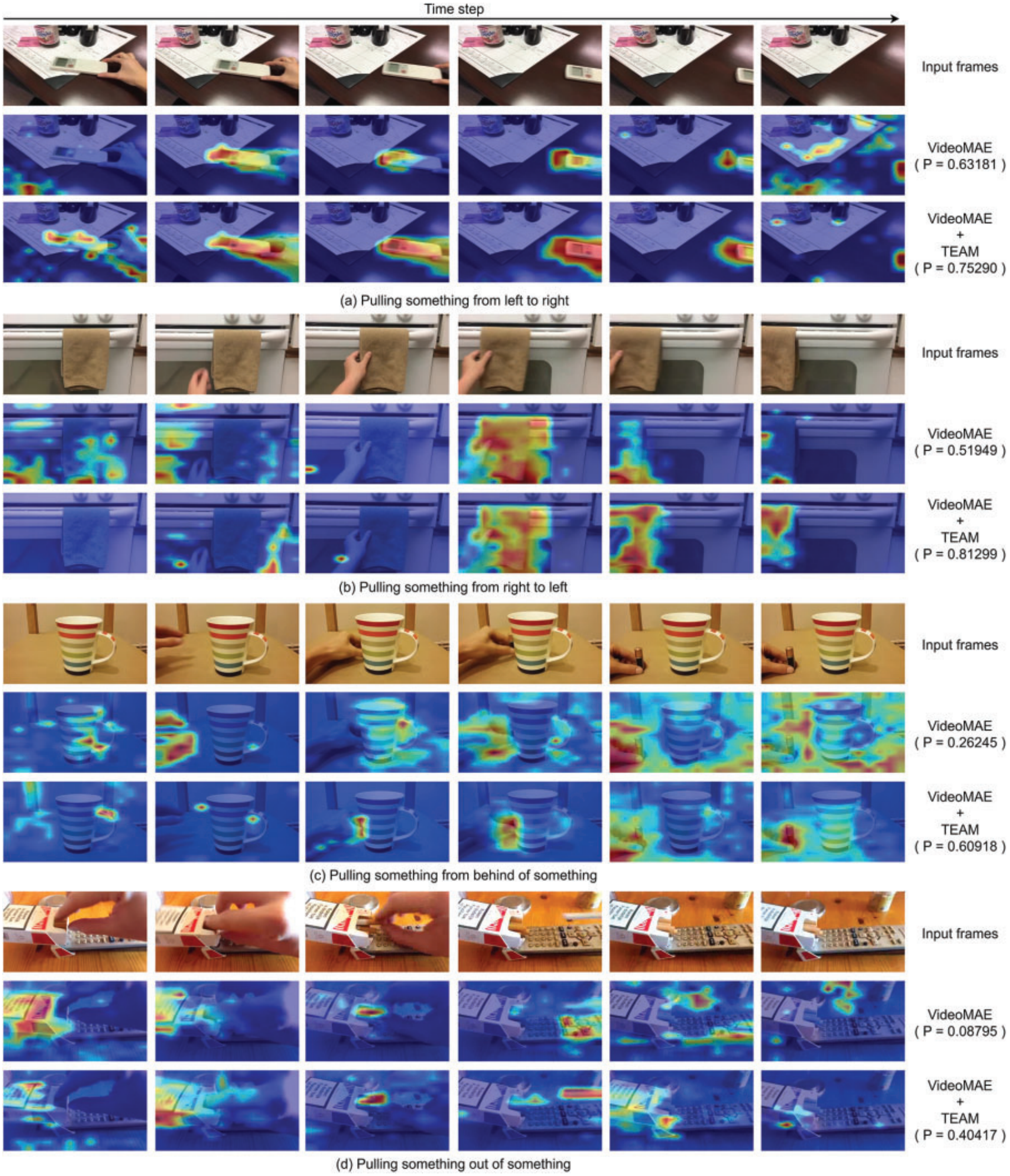

Figure 6: Qualitative comparison of the integrated model, i.e., VideoMAE + TEAM, with the existing VideoMAE on SSv2. We represented visualization results as heat maps, where the red region indicates a high degree of the model's attention. Conversely, the blue region in those heat maps denotes a low degree of the model's attention. All videos in this figure belong to the same action group, i.e., pulling something. P denotes the softmax score of each model for the category

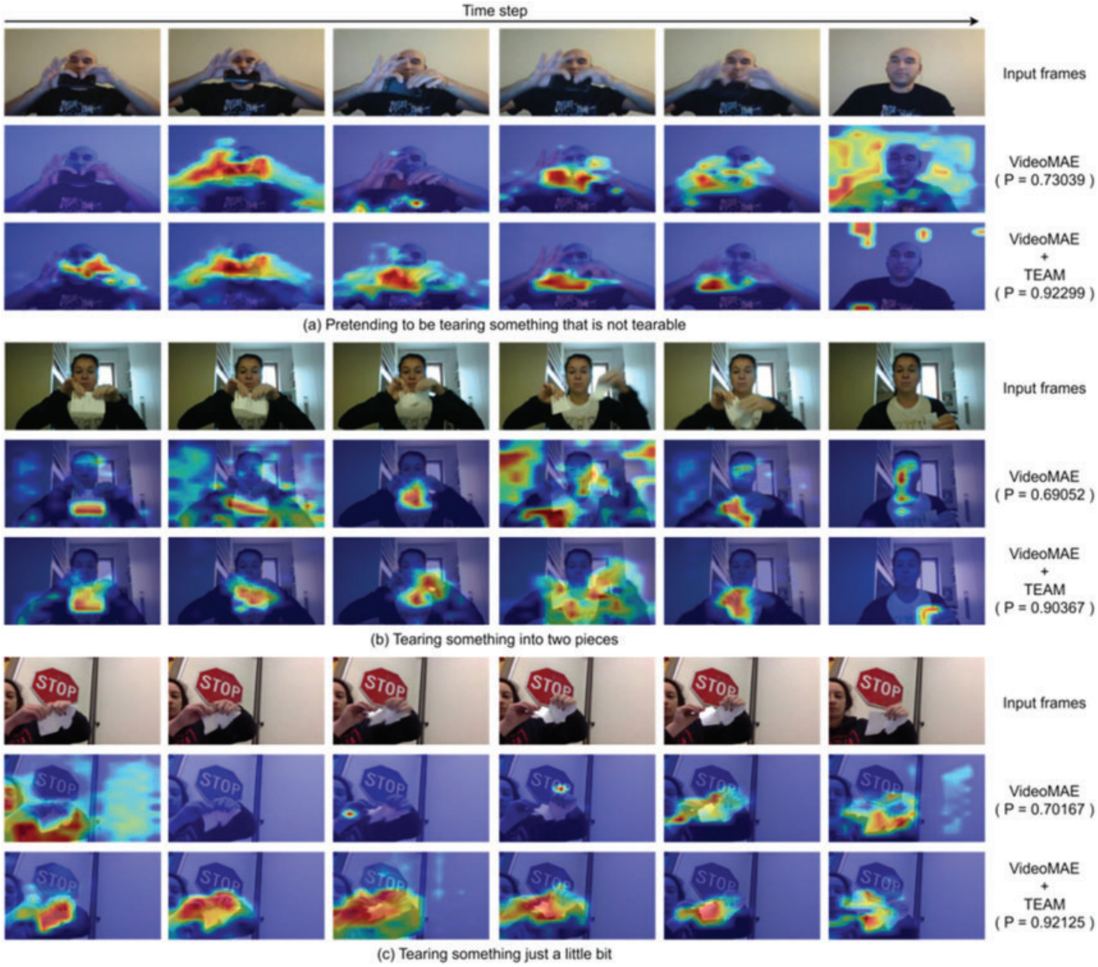
Time step
Input frames
VideoMAE (P = 0.73039)
VideoMAE + TEAM (P = 0.92299)
(a) Pretending to be tearing something that is not tearable
Input frames
VideoMAE (P = 0.69052)
VideoMAE + TEAM (P = 0.90367)
(b) Tearing something into two pieces
Input frames
VideoMAE (P = 0.70167)
VideoMAE + TEAM (P = 0.92125)
(c) Tearing something just a little bit

Figure 7: Qualitative comparison of the integrated model, i.e., VideoMAE + TEAM, with the existing VideoMAE on SSv2. All videos in this figure are from the same action group, i.e., tearing something

5 Conclusion and Future Direction

In this paper, we propose an attention-directing module (i.e., TEAM) that aims to allow the VideoMAE to focus only on the core features in the video. By doing so, it effectively addresses the spatial bias issue, leading to improved performance on video understanding. We first analyze how the VideoMAE processes the contexts of each video. Based on this analysis, we design the FAM which identifies the core features among the overall extracted features. Additionally, to discern the meaningful parts that contain those features, we design the TAM. By sequentially placing the FAM and TAM into the TEAM, the proposed module steers the model's attention from spatial contexts to the main contexts, enabling the model to capture more accurate video contexts. Through extensive experiments, we demonstrate that our design choice allows the TEAM to operate most optimally for its intended purpose. Furthermore, the qualitative results reveal that the TEAM enables the VideoMAE to focus predominantly on motion-related features, resulting in its enhanced ability to capture detailed spatio-temporal contexts in the video.

Even though it has been proven that the TEAM effectively directs the model's attention to the target context of the data, there are still some clear limitations. Firstly, the TEAM has successfully improved performance in SSv2 by seamlessly steering the model's focus to the main spatio-temporal context of the video. However, it is still challenging to achieve significant performance improvements in K400. As mentioned in Section 4.1.2, the target context in K400 is well known to be the spatial context. Therefore, during the training of VideoMAE + TEAM on K400, TEAM adeptly guided the model's focus to the target context as per its intended purpose. However, due to the temporal redundancy, VideoMAE's attention was already directed towards spatial context, limiting the effective impact of the TEAM. Accordingly, we can anticipate that if the TEAM can not only redirect the model's attention to the target context but also refine the captured target context into a more sophisticated form, it would be effective across various datasets. Therefore, we plan to explore the optimal design choice that enhances the model's classification performance by both directing the model's attention and refining the captured target context more precisely.

Secondly, as mentioned in Section 2.2, among several recently proposed SSVP approaches, only the pre-trained weights of VideoMAE are publicly available. Hence, we evaluated the effectiveness of the TEAM by integrating it only with the VideoMAE. Although we have demonstrated that the TEAM is effective for the VideoMAE (ViT-based), it cannot be guaranteed that the TEAM would be equally effective in other SSVP approaches based on different architectures, such as the BEVT (Video Swin Transformer-based) or InternVideo (Uniformer-V2-based). Therefore, in the future, when the pre-trained weights of other SSVP approaches become open-sourced, we plan to evaluate the effectiveness of the TEAM by applying it to those approaches. Furthermore, we will continue to improve the structure of the TEAM and ultimately design a refined attention module that operates effectively across various architectures.

Acknowledgement: The authors would like to thank the National Research Foundation of Korea (NRF), Institute of Information & Communications Technology Planning & Evaluation (IITP), and the Korea government (*MSIT) for funding the publication of this study. *Ministry of Science and Information & Communication Technology.

Funding Statement: This work was supported by the National Research Foundation of Korea (NRF) Grant (Nos. 2018R1A5A7059549, 2020R1A2C1014037) and by Institute of Information & Communications Technology Planning & Evaluation (IITP) Grant (No. 2020-0-01373), funded by the Korea government (*MSIT). *Ministry of Science and Information & Communication Technology.

Author Contributions: The authors' contributions to this study are as follows: study conception and design: Hae Sung Park; data collection: Hae Sung Park; analysis and interpretation of results: Hae Sung Park, Yong Suk Choi; draft manuscript preparation: Hae Sung Park, Yong Suk Choi. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The datasets used and analyzed in this study are publicly available on their official websites. The K400 dataset URLs: (https://www.deepmind.com/open-source/kinetics). The SSv2 dataset URLs: (https://developer.qualcomm.com/software/ai-datasets/something-something).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.,* "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, Long Beach, CA, USA, 2017.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai *et al.,* "An image is worth 16 × 16 words: Transformers for image recognition at scale," arXiv preprint arXiv:2010.11929, 2021.

[3] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lucič *et al.,* "ViViT: A video vision transformer," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision*, Montreal, QC, Canada, pp. 6836–6846, 2021.

[4] H. Bao, L. Dong, S. Piao and F. Wei, "Beit: Bert pre-training of image transformers," arXiv preprint arXiv:2106.08254, 2021.

[5] M. Ding, B. Xiao, N. Codella, P. Luo, J. Wang *et al.,* "Davit: Dual attention vision transformers," in *Computer Vision–ECCV 2022: 17th European Conf.*, Tel Aviv, Israel, pp. 74–92, Springer, 2022.

[6] G. Bertasius, H. Wang and L. Torresani, "Is space-time attention all you need for video understanding?" in *Proc. of the 38th Int. Conf. on Machine Learning*, vol. 139, pp. 813–824, 2021.

[7] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei *et al.,* "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision*, Montreal, QC, Canada, pp. 10012–10022, 2021.

[8] Z. Tong, Y. Song, J. Wang and L. Wang, "Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training," in *Advances in Neural Information Processing Systems*, New Orleans, LA, USA, vol. 35, pp. 10078–10093, 2022.

[9] Y. Wang, K. Li, Y. Li, Y. He, B. Huang *et al.,* "Internvideo: General video foundation models via generative and discriminative learning," arXiv preprint arXiv:2212.03191, 2022.

[10] R. Wang, D. Chen, Z. Wu, Y. Chen, X. Dai *et al.,* "Masked video distillation: Rethinking masked feature modeling for self-supervised video representation learning," arXiv preprint arXiv:2212.04500, 2022.

[11] C. Wei, H. Fan, S. Xie, C. Y. Wu, A. Yuille *et al.,* "Masked feature prediction for self-supervised visual pre-training," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, New Orleans, LA, USA, pp. 14668–14678, 2022.

[12] S. Woo, J. Park, J. Y. Lee and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proc. of the European Conf. on Computer Vision (ECCV)*, Munich, Germany, pp. 3–19, 2018.

[13] Z. Halim, S. Hussain and R. H. Ali, "Identifying content unaware features influencing popularity of videos on youtube: A study based on seven regions," *Expert Systems with Applications*, vol. 206, pp. 117836, 2022.

[14] U. Manzoor and Z. Halim, "Protein encoder: An autoencoder-based ensemble feature selection scheme to predict protein secondary structure," *Expert Systems with Applications*, vol. 213, pp. 119081, 2023.

[15] Uzma, F. Al-Obeidat, A. Tubaishat, B. Shah and Z. Halim, "Gene encoder: A feature selection technique through unsupervised deep learning-based clustering for large gene expression data," *Neural Computing and Applications*, vol. 34, pp. 8309–8331, 2022.

[16] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier *et al.,* "The kinetics human action video dataset," arXiv preprint arXiv:1705.06950, 2017.

[17] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal *et al.,* "The "something something" video database for learning and evaluating visual common sense," in *Proc. of the IEEE Int. Conf. on Computer Vision*, Venice, Italy, pp. 5842–5850, 2017.

[18] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang *et al.,* "Video swin transformer," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, New Orleans, LA, USA, pp. 3202–3211, 2022.

[19] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan *et al.,* "Multiscale vision transformer," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision*, Montreal, QC, Canada, pp. 6824–6835, 2021.

[20] K. Li, Y. Wang, P. Gao, G. Song, Y. Liu *et al.,* "Uniformer: Unified transformer for efficient spatiotemporal representation learning," arXiv preprint arXiv:2201.04676, 2022.

[21] S. Yan, X. Xiong, A. Arnab, Z. Lu, M. Zhang *et al.,* "Multiview transformers for video recognition," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, New Orleans, LA, USA, pp. 3333–3343, 2022.

[22] R. Wang, D. Chen, Z. Wu, Y. Chen, X. Dai *et al.,* "BEVT: Bert pretraining of video transformers," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, New Orleans, LA, USA, pp. 14733–14743, 2022.

[23] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

[24] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh *et al.,* "Grad-CAM: Visual explanations from deep networks via gradientbased localization," in *Proc. of the IEEE Int. Conf. on Computer Vision*, Venice, Italy, pp. 618–626, 2017.

[25] Z. Qing, S. Zhang, Z. Huang, X. Wang, Y. Wang *et al.,* "MAR: Masked autoencoders for efficient action recognition," *IEEE Transactions on Multimedia*, 2023. https://doi.org/10.1109/TMM.2023.3263288

[26] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," arXiv preprint arXiv:1711.05101, 2017.

[27] B. Singh, S. De, Y. Zhang, T. Goldstein and G. Taylor, "Layer-specific adaptive learning rates for deep networks," in *2015 IEEE 14th Int. Conf. on Machine Learning and Applications (ICMLA)*, Miami, FL, USA, pp. 364–368, 2015.

[28] R. Müller, S. Kornblith and G. E. Hinton, "When does label smoothing help?" in *Advances in Neural Information Processing Systems*, vol. 32, Vancouver, BC, Canada, 2019.

[29] H. Zhang, M. Cisse, Y. N. Dauphin and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," arXiv preprint arXiv:1710.09412, 2017.

[30] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe *et al.,* "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision*, Seoul, Korea (South), pp. 6023–6032, 2019.

Appendix A. Training Details

As mentioned in Section 4.1.3, except for the batch size and learning rate, we kept all training settings consistent with the settings used for training the existing VideoMAE. All the training settings for the integrated model, i.e., VideoMAE + TEAM, are presented in Table 10. Specifically, we randomly initialized all the learnable parameters of the TEAM and applied the TEAM to the VideoMAE for training.

Table 10: Training configuration for experiments on Kinetics-400 and Something-Something-V2. Here, B and S refer to the integrated model, namely VideoMAE-Base + TEAM and VideoMAE-small + TEAM, respectively

Training configuration	Dataset	
	Kinetics-400	Something-Something-V2
Optimizer	AdamW [26]	
Base learning rate	2.5e-4 (S) 1.768e-4 (B)	
Weight decay	0.05	
Layer–wise lr decay [27]	0.75	
Batch size	24 (S) 12 (B)	
Learning rate schedule	Cosine decay	
Warmup epochs	5	
Training epochs	150 (S) 100 (B)	40 (S) 30 (B)
Flip augmentation	✓	×

(Continued)

Table 10 (continued)

Training configuration	Dataset	
	Kinetics-400	Something-Something-V2
Label smoothing [28]	0.1	
Mix up [29]	0.8	
Cutmix [30]	1.0	
Drop path	0.1	