This CVPR Workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Effectiveness of Max-Pooling for Fine-Tuning CLIP on Videos

Fatimah Zohra Chen Zhao Shuming Liu Bernard Ghanem
King Abdullah University of Science and Technology (KAUST)

Abstract

CLIP is a powerful spatial feature extractor trained on a large dataset of image-text pairs. It exhibits strong generalization when extended to other domains and modalities. However, its extension to videos is challenged by the need for additional temporal modeling. While recent works have attempted to bridge this modality gap through the integration of complex modules, we apply a simple and modular approach to enhance CLIP's video understanding on action recognition tasks. In its standard application, CLIP processes each video frame independently, restricting its ability to associate features across frames. To address this, we apply frame-wise max-pooling on the tokens within the transformer layer to construct a new set of tokens that aid the model in extracting temporal information better. We then use max-pooling to aggregate the features into a single video feature. We evaluate the effectiveness of this approach on different action recognition benchmarks, showing that max-pooling is able to help fine-tune the model to extract the features important for temporal modeling. Furthermore, we show that the max-pooling of tokens is particularly useful when applied to the last few layers of the model, which are typically more specialized and abstract for capturing high-level image features. To the best of our knowledge, we are able to achieve SOTA on the base-to-novel and few-shot benchmarks on the Something-SomethingV2 dataset.

1. Introduction

Large multi-modal image-text models [14] [49] [10] have shown promising potential for video tasks – CLIP [31], in particular, has seen many applications to the video domain. CLIP comprises a transformer-based image encoder [8] and text encoder [37] which encodes the image and its corresponding text independently. A contrastive loss based on a similarity measure is used to align these two modalities [36]. For action recognition, many works leverage the spatial features learned by CLIP, further fine-tuning the model to extract temporal features. These fine-tuning approaches tend to introduce complex inductive biases to extract the temporal features from videos. [26] sequentially inserts additional cross-frame attention modules with frame-level and video-level token exchange in order to facilitate the flow of temporal information. In contrast, [24] inserts cross-frame modules (applying either temporal attention or a 3D convolution) in a parallel branch structure. [20] proposes a more complex design altogether, dubbed the Uniformer block, which unifies the self-attention mechanism with convolutions to aggregate both frame-level and video-level features. More light-weight temporal feature extraction models aim to expand the temporal view within the spatial self-attention of the Transformer by frame-wise shifting the channels of the tokens [40] or frame-wise alignment of tokens to gather temporal information [35].

Alternatively, parameter-efficient fine-tuning approaches freeze the CLIP model as a spatial feature extractor and introduce either temporal adaptation layers [27] [48] or additional temporal encoders [30] and temporal decoders [23]. [13] uses a feature distillation approach, treating the frozen CLIP model as a teacher while fine-tuning a student model followed by a residual projection layer. Such designs tend to extract spatial and temporal features either sequentially or in parallel with each other, often relying on an integration mechanism or some form of token exchange in or across the Transformer blocks for fusing these two aspects of the features together.

Despite these intricate designs targeting the learning of temporal features, [32] shows that a simple frame-level feature aggregation using mean-pooling is a competitive baseline to bridge the modality gap from images to videos when fine-tuning CLIP on several action recognition benchmarks. [32] additionally shows that prompting both the image and text encoder helps improve the performance on several such tasks.

In addition to the inductive biases introduced by these methods, many continue to employ some aggregation method to aggregate the frame-level features into a video-level feature – the simplest being an average-pooling of the frame-level features, to more parametrically complex aggregators including 1D-convolution layers, LSTMs, Transformers [25] [40], and adaptive-pooling, which adds an integration branch between the spatial and temporal features to fuse them together [30]. In the case of [20], the video-

level tokens from different stages of the model are fused together either sequentially, in parallel, or hierarchically to obtain a comprehensive representation.

Our approach investigates the potential of max-pooling to effectively aggregate the frame-level features into a video-level feature. We ask the following question: *can we apply max-pooling in a specific way that nudges the model to focus on areas that show temporal change?* Surprisingly, max-pooling in and of itself consistently outperforms the widely used average-pooling across different benchmarks. In combination with additionally max-pooling the tokens in the last few layers of CLIP's image encoder, the model's ability to generalize significantly improves.

Our contributions are as follows:

- We show the effectiveness of max-pooling for aggregating frame-level features into a video-level feature and propose this to be a stronger baseline for temporal aggregation when fine-tuning CLIP on videos. To the best of our knowledge, models applying CLIP to video data have relied on mean-pooling and other complex late-fusion strategies despite the apparent benefits of the max-pooling operation we observe.
- We additionally propose a zero-parameter, lightweight, and modular approach for improving the temporal modeling by injecting frame-wise max-pooled tokens after self-attention in CLIP's image encoder. We observe that max-pooling of the features alone has the potential for overfitting and that additionally pooling the tokens as such significantly improves generalization.
- For the action recognition task, we conduct experiments on three different evaluation settings, base-to-novel, few-shot, and fully supervised, and four datasets, Kinetics-400 [3], Something Something V2 [12], HMDB51 [19], and UCF101 [34]. Our approach is able to achieve competitive performance on several of these benchmarks as well as SOTA on the Something Something V2 dataset in the base-to-novel and few-shot settings.

2. Related Work

2.1. CLIP for Vision-Language Representation Learning

CLIP [31] has demonstrated impressive zero-shot performance on several computer vision tasks, not only in the image domain [52], [39] [43], but also for various video understanding [47] [40] [15] [26] [30] [35] and video-language understanding tasks [25] [9] [2] [18] among others. Its ability to match images with text descriptions without having been trained on the full distribution of image-text pairs confers a very desirable semantic space for both the open-vocabulary [44] [32] and few-shot settings [50] [11] [53] [32] [22] in particular. In the open-vocabulary setting, the model trains on base classes and inferences on both base and novel classes [45]. It is notably distinct from the typical zero-shot setting in that the model can classify novel classes with the help of a large language vocabulary, whereas in the zero-shot setting, the model must identify novel classes solely based on textual cues. In the video domain, [32] proposes an open-vocabulary base-to-novel setting that is used for the evaluation of many SOTA CLIP-based models.

2.2. CLIP for Action Recognition

The task of action recognition in videos is characteristically marked by not only having to extract spatial features, but also the motion in the regions of the images across frames. The extraction of these temporal features is crucial for strong performance, especially on video datasets like Something Something V2 [12], which cannot be solved as easily by exploiting a spatial bias alone. While the pretraining on image-text pairs make CLIP inherently less suited to video-text fine-tuning for action recognition in some respect – consequently leading to advances in video-text foundational model [1] [42] (some of which use the frozen CLIP model [21]), several works have nevertheless effectively leveraged CLIP's zero-shot capabilities in the video domain [32] [40], [13], [44] [22] [6] [46] [35] [48] [27].

In the extension of CLIP to videos, some methods seeking primarily to evaluate action recognition tend to discard the text encoder. However, as [32] shows, jointly training both the image and text encoder such that the action classes are embedded into prompts results in a superior baseline performance, especially notable in generalization settings such as zero-shot, few-shot, and base-to-novel. Furthermore, recent works have shown that qualitatively augmenting the video-text pairs using approaches such as video-conditioned text [16] or LLM-generated text descriptions [22] [13] [5] [7] is beneficial. In our work, we also fine-tune CLIP's pretrained text encoder.

2.3. Expanding the Temporal View of Tokens

While many of the modifications extending CLIP to videos tend to introduce new components for temporal modeling [26] [24] [30] [20] [23], few methods have tried to impart a light-weight inductive bias for learning the temporal dynamics. [35] aligns the features of adjacent frames using a learned mask over the image features. It uses a convolution module to predict semantically similar positions between the frames. The predicted region in turn is used to weight the frame-level features. The resulting aligned features are mean-pooled into a single token, which is concatenated with the corresponding frames before being transformed by the Transformer's self-attention.

Also notable is [40], which was among the earliest applications of CLIP to the action recognition task. [40] relied on CLIP's image encoder to process each frame independently, but with the added caveat of shifting the channels of the to-

kens across frames. In doing so, each token, which in the standard Transformer block corresponds to a spatial *patch* of the input frame, was modified into a combination of the adjacent frames. This stacking of the channels from multiple frames expands the temporal view of each token, and subsequently of the otherwise spatial self-attention block. In this way, the fully connected layers, which are designed to process each token independently, receive an input spanning multiple frames.

In a similar fashion, our approach also targets the tokens directly in order to try to expand the temporal view available to the fully connected layers. However, while both these approaches elicit information from adjacent frames, our approach combines information from the full gamut of frames via a simple max-pooling of the tokens along the temporal dimension. This simple change outperforms both the channel shifting and the deep prompting approaches in the base-to-novel and few-shot settings for the SSv2 [12] dataset.

3. Method

Our goal is to evaluate the effectiveness of max-pooling to extract temporal features when fine-tuning CLIP on videos. To this extent we make two primary modifications to the adopted baseline of CLIP [32], which jointly trains both the image and text encoder by processing frames independently while applying a late fusion of image features using mean-pooling. We first introduce a standalone module in the Transformer for injecting the max-pooled set of tokens in the layers of the image encoder. Secondly, we aggregate the resulting features into a single video feature using max-pooling.

3.1. Max-pooling Tokens

Given a video $V_i \in \mathbb{R}^{T\times 3\times H\times W}$ with T sampled frames and a text label y_i, each frame is independently encoded by the image encoder $f_{\theta^v}(V_i)$ and each text label is wrapped in a prompt template such as '`a photo of a <category>`', which is tokenized and then encoded by the text encoder $t_i = f_{\theta^t}(tokenize(y_i))$. The image encoder consists of a patchification operation on the input to transform each frame into a fixed number of tokens, N, with a fixed channel dimension D. These tokens are then processed by the Transformer layers, each comprising a multi-head self-attention and a two-layer fully-connected network with pre-normalization. Our approach inserts max-pooling between self-attention and a fully connected network, aggregating the tokens across the temporal dimension. Given the output of self-attention $z_i^\ell \in \mathbb{R}^{N\times T\times D}$, the max-pooling operation is applied along the temporal dimension $\hat{z}_i^\ell = \max_{\text{dim}=2}(z_i^\ell)$ where $\hat{z}_i^\ell \in \mathbb{R}^{N\times 1\times D}$ and $\ell \in L$ is the layer of a transformer with L layers. This operation retains the maximum feature value across all time frames while compressing the input into a single representative frame, yielding a new set of N tokens.

The pooled tensor, $\hat{z}_i^\ell$, is then concatenated frame-wise with the existing tokens to get $\hat{z}_i^\ell \in \mathbb{R}^{N\times T+1\times D}$, effectively augments the predominantly spatial tokens with a global temporal context for each token. As such, this combination helps extract temporal features relevant to understand the entire video, rather than the spatial features relevant to an individual frame alone.

The pooled tokens injected at each layer then continue to be processed by subsequent layers. However, instead of applying the pooling at all layers, it is more effective to apply it to the last few layers. Thus, a hyper-parameter $s \leq L$ can be used to specify how many of the last s layers of the L layers of the transformer to max-pool the tokens at. The accrued output of Transformer after all L layers is $\hat{z}_i^L \in \mathbb{R}^{N\times T+s\times D}$. This is then projected by the CLIP image encoder to a fixed embedding dimension E to obtain an output $\hat{z}_i \in \mathbb{R}^{T+s\times E}$

3.2. Max-pooling Features

The output features of the image encoder are once again max-pooled across the temporal dimension to obtain a final video-level feature, $v_i = \max_{\text{dim}=1}(\hat{z}_i)$ where $v_i \in \mathbb{R}^E$. This application of max-pooling is intended to capture the most prominent features from the output of the Transformer layers, further highlighting areas of significant change in the video, under the assumption that these areas are likely to have higher feature values. In the context of action recognition, the relevant information is often contained in these dynamically changing parts of the video, which we hypothesize is the reason for the effectiveness of this approach.

In order to align the video-level features with the corresponding text embedding, we compute the similarity of v_i and t_i using the dot product and then maximize the similarity scores for matching video-text pairs using the cross-entropy objective, as is the standard for training CLIP in the video domain.

4. Experiments

We evaluate our approach on two generalization-specific settings, base-to-novel and few-shot learning, in addition to the fully-supervised setting.

4.1. Comparison to State of the Art

We compare our method to models aimed at applying CLIP to the task of action recognition. We additionally compare it to [32] using max-pooling in lieu of mean-pooling to aggregate the features, denoted as VIFI-CLIP-max or with mean-pooling to aggregate the features, but max-pooling of the tokens in the Transformer image encoder.

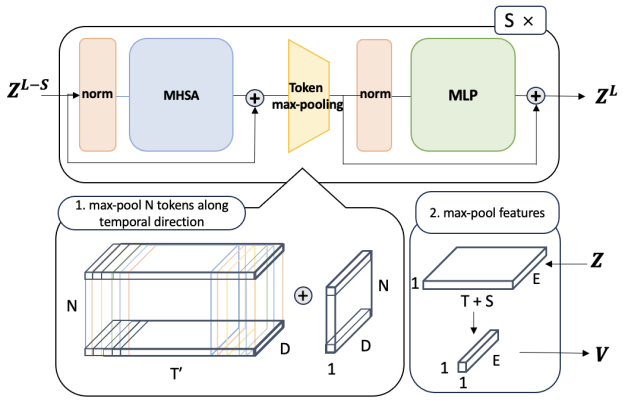

Figure 1. The figure illustrates our proposed approach, which is conceptually quite simple. Given a Transformer with L layers, the Token max-pooling operation is inserted into the last $S \leq L$ layers. (1) In each of the S layers, the token max-pooling operation is applied *after* the self-attention layer. Given the output of the first $L-S$ standard Transformer layers, $Z^{L-S} \in \mathbb{R}^{N\times T\times D}$, our proposed Transformer layer max-pools the tensor along the temporal dimension. The pooled tensor, $\hat{Z}^{L-S} \in \mathbb{R}^{N\times 1\times D}$ is concatenated with the existing tensor to get $\hat{Z}^{L-S} \in \mathbb{R}^{N\times T\prime\times D}$ where $T\prime = T + L - S + 1$ for each successive block S in the remaining layers of the Transformer. Thus, after the S layers, the output is $Z^L \in \mathbb{R}^{N\times(T+S)\times D}$. (2) The output of CLIP's image encoder, which projects the features into a fixed E dimensional embedding, is max-pooled once again to obtain the final video-level feature V.

Base-to-novel setting. The application of max-pooling to aggregate a video-level context gives a significant increase on the evaluated benchmarks – achieving the second-best top-1 accuracy, 18.2, on the base evaluation and the best top-1 accuracy 13.3 on the novel evaluation for the SSv2 dataset, as shown in Table 1. In comparison to our baseline [32], we also observe a notably larger relative increase in performance on the SSv2 dataset, from 16.2 to 18.2 top-1 accuracy (+2) on the base classes. Whereas on the K-400 dataset, this relative increase (+1.1) is smaller, from 76.4 to 77.6 top-1 accuracy on the base classes. This difference suggests the effectiveness of max-pooling in targeting the extraction of temporal features specifically since action recognition in the SSv2 dataset necessitates stronger temporal modeling. The performance of our models on the novel split for the K-400 dataset, however, notably decreases compared to the closely related baseline of [32] at 61.1. Throughout our experiments, this is the only significant case in which the application of max-pooling reduces the baseline performance. In our ablations (see Table 4), we observe that on the K-400 novel split, the performance increases (+0.4 from 61.1 to 61.5) if we instead apply a combination of mean-pooling with the max-pooling of the tokens.

Few-shot setting. In this setting, we evaluate the SSv2, HMDB51, and UCF101 datasets. The most notable increase in the top 1 accuracy is when $K = 16$, in which we observe an increase of +1, from 13.5 to 14.5 as shown in Table 2. Interestingly, it outperforms [22], which additionally pretrains on the Kinetics-400 data.

Fully-supervised setting We evaluate the performance of our models on the K-400 dataset. We observe that applying max-pooling exclusively yields a better performance than the closest baseline [32] (+0.2 from 83.9 to 84.1). However, additionally max-pooling the tokens does not help improve the performance in this case. We speculate that the observed performance may be explained by having trained the model on 16 frames since, as shown in Table 10, our model performs better with more frames.

4.2. Ablations

We evaluate the effectiveness of max-pooling depending on several key design choices for the model's architecture and training. Given that [32] is the closest baseline to our experiments in that it applies a simple mean-pooling of the frame-level features, we closely compare our model with this baseline.

Max-pooling for Temporal Modelling
As shown in Table 4, max-pooling the tokens yields a significant increase regardless of the choice of late-fusion to aggregate features using either mean or max pooling of the features across the base to novel and few-shot settings. The exception is the decrease in performance of the K-400 dataset [3] on the novel classes as discussed in Section 4.1. Furthermore, the harmonic mean, which is a conservative estimate of the average, tends to be highest for both types of frame-level aggregators (mean and max) when the tokens are additionally max-pooled. Despite the clear increase in performance on the base classes, the notable drop in performance (-1.2) on the novel classes when max-pooling on the features exclusively (from 12.1 to 11.3) in the base-to-novel evaluation of SSv2 suggests that max-pooling alone as an aggregation strategy may be vulnerable to overfitting. In the open-vocabulary setting, which necessitates stronger generalization, the additional max-pooling of tokens is beneficial.

Mean-Pooling Features with Max-pooling Tokens. As Table 5 shows, when the frame-level features are averaged, the token max-pooling needs to be applied at earlier layers (best top-1 accuracy is when it is applied to all 12 layers of the transformer when $\text{mean}_{s=12}$ achieves 16.84 on the base classes and 12.77 on the novel classes in the base to novel setting. In contrast, our approach reduces the need to max-pool in earlier layers of the image encoder's Transformer showing that the model benefits best when the token pooling targets only the later layers.

Table 1. **Base-to-novel setting:** Comparison of the top-1 accuracy of our approach with other CLIP-based action recognition models evaluated on the K-400 [3] and SSv2 [4] datasets. s indicates the layer at which the max-pooling of tokens starts applying. HM denotes the harmonic mean on the base and novel performance. ††indicates models trained with LLM based text augmentations. We gray out the results for models for fair comparison. Our model are the first results reported for the base-to-novel setting using a ViT-L/14. † denotes that the evaluation was done by [13]. ‡denotes deep prompting in the image and text encoder.

	K-400			SSv2		
Model	Base	Novel	HM	Base	Novel	HM
Vanilla CLIP [31]	62.3	53.4	57.5	4.9	5.3	5.1
ActionCLIP [40]	61.0	46.2	52.6	13.3	10.1	11.5
XCLIP [26]	74.1	56.4	64.0	8.5	6.6	7.4
VPT-A5 [15]	69.7	37.6	48.8	8.3	5.3	6.4
AIM† [48]	74.6	62.5	68.0	8.5	7.9	8.2
ST-Adaptor† [27]	73.6	62.0	67.3	9.3	8.4	8.8
ViFi-CLIP ‡ [32]	-	-	-	15.8	11.5	13.3
VILT-CLIP ‡ [38]	77.4	63.0	69.5	17.3	12.8	14.7
EZ-CLIP [33]	73.1	60.6	66.3	16.6	13.3	14.8
TC-CLIP [17]	**78.9**	**63.6**	**70.4**	17.5	**13.4**	15.2
Froster †† [13]	77.8	64.3	70.4	18.3	12.2	14.6
ViFi-CLIP [32]	76.4	61.1	67.9	16.2	12.1	13.9
ViFi-CLIP-max	77.6	58.6	66.8	17.4	11.3	13.7
ViFi-CLIP (w/ max-pooling tokens only)	76.4	61.5	68.2	16.8	12.8	14.5
ours$_{s=1}$-B/16	77.6	58.1	66.5	**18.2**	13.3	**15.4**
ours$_{s=8}$-L/14	83.1	67.1	74.3	23.1	16.1	19.0

Table 2. **Few-shot setting:** Comparison of the top-1 accuracy of our approach with other CLIP-based ViT-B/16 action recognition models evaluated on the SSv2 dataset. †indicates the model is pretrained on the Kinetics-400 dataset and hence is not a fair comparison. ‡denotes deep prompting in the image and text encoder. ††indicates models trained with LLM based text augmentations. We gray out the results for fair comparison. * indicates s=4 for the SSv2 dataset and s=3 for the HMDB51 and UCF101 datasets.

Model	SSv2				HMDB51				UCF101			
	K=2	K=4	K=8	K=16	K=2	K=4	K=8	K=16	K=2	K=4	K=8	K=16
Vanilla CLIP [31]	2.7	2.7	2.7	2.7	41.9	41.9	41.9	41.9	63.6	63.6	63.6	63.6
ActionCLIP [40]	4.1	5.8	8.4	11.1	47.5	57.9	57.3	59.1	70.6	71.5	73.0	91.4
XCLIP [26]	3.9	4.5	6.8	10.0	53.0	57.3	62.8	64.0	48.5	75.6	83.7	91.4
VPT-A5 [15]	4.4	5.1	6.1	9.7	39.7	50.7	56.0	62.4	71.4	79.9	85.7	89.9
ASU [6]	-	-	-	-	60.1	63.8	67.2	70.8	**91.4**	**94.6**	**96.0**	**97.2**
EZ-CLIP [33]	6.8	8.7	9.6	13.2	57.3	61.1	65.4	67.7	84.7	88.3	91.3	92.8
TC-CLIP [17]	7.3	8.6	9.3	14.0	57.3	62.3	67.3	68.6	85.9	89.9	92.5	94.6
ViFi-CLIP ‡[32]	6.7	7.9	10.2	13.5	**63.0**	**65.1**	**69.6**	**72.0**	91.0	93.7	95.0	96.4
VILT-CLIP ‡[38]	**7.8**	**9.4**	**10.3**	13.2	60.6	61.9	66.9	69.6	85.3	90.0	91.3	93.8
VicTR† [16]	-	-	-	-	60.0	63.2	66.6	70.7	87.7	92.3	93.6	95.8
MAXI † [22]	7.1	8.4	9.3	12.4	58.0	60.1	65.0	66.5	86.8	89.3	92.4	93.5
ALT †† [7]	-	-	-	-	64.3	66.7	70.4	74.5	93.2	95.3	96.4	97.3
OST †† [5]	7.0	7.7	8.9	12.2	59.1	62.9	64.9	68.2	82.5	87.5	91.7	93.9
ViFi-CLIP [32]	6.2	7.4	8.5	12.4	57.2	62.7	64.5	66.8	80.7	85.1	90.0	92.7
ViFi-CLIP-max	7.1	8.5	10.0	**14.5**	58.9	60.5	63.9	65.8	81.0	86.7	90.0	92.8
ours$_{s=3}$-B/16	7.2	8.5	10.0	**14.5**	58.9	61.9	64.2	66.7	80.7	86.7	90.4	93.1
ours$_{s=4/3}$-L/14*	9.8	11.6	13.8	18.1	62.1	63.8	71.1	73.0	88.7	92.1	94.8	95.9

Significance in Later Layers. As shown in Tables 6 and 7, the performance of our approach improves when the max-pooling of the tokens is applied at later layers of the Transformer. We conjecture that this may be because the initial layers tend to capture more fine-grained features. By avoiding max-pooling of the tokens in these early layers, the

Table 3. **Fully-supervised setting:** Comparison with CLIP-based models trained on Kinetics-400 using the ViT-B/16 architecture. ‡indicates that the model is additionally post-pretrained on the K710-0.66M dataset. ††indicates models trained with LLM based text augmentations.

Model	Frames	Views	K400 (Top-1)	K400 (Top-5)
Parameter Efficient Fine-tuning				
VILT-CLIP [38]	16	4x3	77.6	94.5
M^2CLIP [41]	16	4x3	83.7	96.7
MoTED †† [51]	16	3x1	85.4	97.2
Qian et al. [29]	16	3x1	**86.1**	**97.5**
ST-Adaptor [27]	32	3x1	82.7	96.2
AIM [48]	32	3x1	84.9	96.7
EVL [23]	32	1x3	84.2	-
DUALPATH [28]	32	3x1	85.4	97.1
Full Fine-tuning				
UniformerV2 ‡ [20]	8	3x4	85.6	97.0
DIST [30]	32	1x3	85.0	97.0
ActionCLIP [40]	32	10x3	83.8	96.2
XCLIP [26]	16	4x3	84.7	96.8
VPT-A6 [15]	16	-	76.9	93.5
VicTR [16]	16	4x3	84.2	-
ASU [6]	16	4x3	84.8	96.7
ILA [35]	16	4x3	<u>85.7</u>	<u>97.2</u>
TC-CLIP [17]	16	4x3	85.2	96.9
ALT [7]	16	3x1	84.8	96.4
ViFi-CLIP [32]	16	4x3	83.9	96.3
ViFi-CLIP-max	16	4x3	84.1	96.6
ours$_{s=3}$	16	4x3	83.9	96.2

model can maintain a higher resolution of the low-level spatial context for each frame. As the input progresses through the model's layers and the extracted features become more high-level, applying max-pooling in the later layers allows the model to aggregate these high-level features across the temporal dimension.

Position of Max-Pooling of Tokens. We observe that max-pooling the tokens *after* self-attention is slightly more effective than max-pooling the tokens *before* self-attention. In the Transformer, self-attention is designed to weigh the importance of different tokens relative to each other. Since the max-pooling operation aims to extract features across the frames, it is better positioned after the self-attention so that the model can learn a global context from the most significant spatial features determined by the attention mechanism. We empirically validate this in Table 8 and Table 9 for the base to novel and few shot settings.

16 vs 32 Frames. For the most part, we observe that our approach tends to improve with more frames as shown in Table 10 and Table 11. This is more notable for the Kinetics-400 dataset [3]. This difference suggests that the more variability there is in the feature values, the more discerning the max-pooling-based temporal modeling becomes.

ViT-B/16 vs ViT-L/14. Interestingly, we observe that max-pooling as a late fusion step results in greater performance in the larger ViT-L/14 models. As shown in Table 12, the top 1 accuracy on the base-to-novel increases by +2.6, from 20.3 when using mean-pooling to aggregate the features to 22.9 when using max-pooling to aggregate the features. Furthermore, with the additional max-pooling of tokens applied, we observe an additional increase in performance to 23.1 top-1 accuracy on the base set and 16.1 top-1 accuracy on the novel set. We note that while max-pooling the tokens in earlier layers does not hurt the performance on the base classes (max$_{s=24}$), it decreases the performance on the novel classes (from 16.0 to 14.8). However, max-pooling the tokens for half the layers of the transformer (max$_{s=12}$) not only yields strong performance on the bases classes, 23.0, but also on the novel classes, 16.1. In comparison, a ViT-B/16 sees an increase in the top 1 accuracy by +1.2, from 16.2 when using mean-pooling to aggregate the features to 17.4 when using max-pooling to aggregate the features. Table 13 shows the results for the few-shot setting, which also gives higher performance than the ViT-B/16 counterparts.

Table 4. **Mean-Pooling vs Max-Pooling of Features and Tokens**. Comparison of the top 1 accuracy on using a combination of mean-pooling or max-pooling to aggregate the features into a video representation, denoted as *mean* or *max* respectively. Each model is optionally trained with or without the max-pooling of tokens in the last s layers of the Transformer, denoted by the subscript (ex. $s = 1$ denotes the last layer). The default settings for each model use 32 frames and apply the token max-pooling *after* self-attention in the Transformer layer. † denotes the results reported by [32].

SSv2			
Method	Base	Novel	HM
mean †	16.2	12.1	13.9
mean$_{s=12}$	16.8	12.8	14.5
max	17.4	11.3	13.7
max$_{s=1}$	18.2	13.3	15.4

K-400			
Method	Base	Novel	HM
mean†	76.4	61.1	67.9
mean$_{s=12}$	76.4	61.5	68.2
max	77.6	58.6	66.8
max$_{s=1}$	77.6	58.1	66.5

SSv2				
Method	K=2	K=4	K=8	K=16
mean†	6.2	7.4	8.5	12.4
mean$_{s=12}$	6.9	7.8	9.3	13.0
max	7.1	8.5	10.0	14.5
max$_{s=3}$	7.2	8.5	10.0	14.5

Table 5. **Mean-Pooling of Features** Comparison of the top 1 accuracy of our max-pooling token approach for different layers s of the Transformer when aggregating the features using mean-pooling.

SSv2			
Method	Base	Novel	HM
mean$_{s=12}$	16.84	12.77	14.53
mean$_{s=9}$	16.78	12.57	14.37
mean$_{s=6}$	16.59	12.48	14.24
mean$_{s=3}$	16.49	12.11	13.96
mean$_{s=1}$	16.40	12.01	13.87

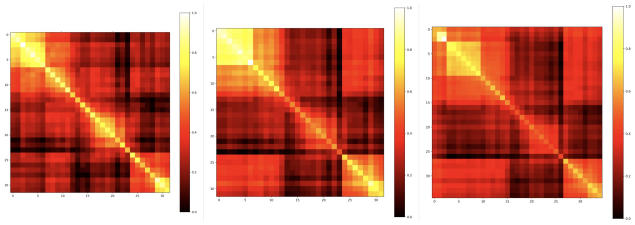

Figure 2. These plots visualize the cosine similarity matrix of video features, where each cell represents the dot product similarity between pairs of temporal features for a video. Brighter values indicate greater similarity. The matrices are derived by averaging the individual matrices across all videos in the validation dataset. They correspond to different approaches, mean-pooling of features, max-pooling of features, and max-pooling of both tokens and features respectively in the few-shot $k = 16$ setting using a ViT-L/14 model on the SSv2 dataset. We observe that these trends hold in general across the different evaluation settings and datasets.

5. Discussion

We reason that max-pooling improves the model's ability to aggregate many distinct motion patterns extracted across multiple frames and map them to a stationary concept of movement useful for the task of action recognition. To analyze this, we employ a cosine similarity matrix of the frame-level features across the videos. As shown in figure 2, we observe that when the model is trained with mean-pooling of the frame-level features, there is greater similarity in the learned features, indicated by the larger and brighter clusters along the diagonal. We interpret these clusters as segments in the video in which the action stays relatively consistent over several frames. For the task of action recognition this similarity leads to more redundancy in the temporally aggregated feature. In contrast, with max-pooling, we observe that the brightness along the diagonal is less intense and there is additionally increased density of dark regions. This reveals that the adjacent frame-level features capture more distinct features, thereby helping to preserve the nuances in temporal information (i.e. the motion in the video) during the aggregation.

6. Conclusion

The experiments demonstrate the effectiveness of max-pooling of tokens and features to different degrees on different benchmarks. Despite these differences, the application of max-pooling generally improves the performance compared to the mean-pooling baseline widely used for temporal aggregation of frame-level features into a video-level representation.

Table 6. Significance of Later Layers: Base-to-novel setting

	SSv2		
Method	Base	Novel	HM
$\max_{s=6}$	17.7	12.8	14.9
$\max_{s=3}$	18.0	12.6	14.8
$\max_{s=1}$	18.2	13.3	15.4

Table 7. Significance of Later Layers: Few-shot setting

Model	SSv2			
	K=2	K=4	K=8	K=16
$\max_{s=6}$	7.1	8.6	9.6	14.3
$\max_{s=3}$	7.2	8.5	10.0	14.5
$\max_{s=1}$	7.0	8.6	10.0	14.3

Table 8. Position of Token Max-Pooling: Base-to-novel setting

	K-400	SSv2
Method	Base	Base
$\max_{s=3,before}$	77.4	17.8
$\max_{s=3,after}$	77.5	18.0

Table 9. Position of Token Max-Pooling: Few-shot setting

Model	SSv2			
	K=2	K=4	K=8	K=16
$\max_{s=3,before}$	7.2	8.4	10.0	14.4
$\max_{s=3,after}$	7.2	8.5	10.0	14.5
$\max_{s=1,before}$	6.8	8.5	9.8	14.2
$\max_{s=1,after}$	7.0	8.6	10.0	14.3

Table 10. 16 vs 32 Frames: Base-to-novel setting

	K-400	SSv2
Method	Base	Base
$\max_{s=1,f=16}$	76.8	18.0
$\max_{s=1,f=32}$	77.6	18.2

Table 11. 16 vs. 32 Frames: Few-shot setting

Model	SSv2			
	K=2	K=4	K=8	K=16
$\max_{s=3,f=16}$	7.0	8.5	10.3	14.4
$\max_{s=3,f=32}$	7.2	8.5	10.0	14.5

Table 12. ViT-L/14: Base-to-novel setting

	SSv2		
Method	Base	Novel	HM
mean	20.3	15.3	17.6
max	22.9	16.0	18.8
$\max_{s=24}$	23.0	14.8	18.0
$\max_{s=12}$	23.0	16.2	19.0
$\max_{s=8}$	23.1	16.1	19.0
$\max_{s=3}$	23.0	16.0	18.9

Table 13. ViT-L/14: Few-shot setting

Model	SSv2			
	K=2	K=4	K=8	K=16
mean	9.1	10.3	12.1	15.2
max	9.6	11.4	13.6	18.0
$\max_{s=3}$	9.7	11.6	13.6	18.0
$\max_{s=4}$	9.8	11.6	13.8	18.1

Table 14. Comparison of the top 1 accuracy on using a combination of mean-pooling or max-pooling of features, denoted as **mean** or **max** respectively. Each model is additionally trained with or without the max-pooling of tokens in the last s layers of the Transformer, denoted by the subscript (ex. $s = 1$ denotes the last layer). The default settings for each model use 32 frames and apply the token max-pooling *after* self-attention in the Transformer layer.

References

[1] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[2] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. A clip-hitchhiker's guide to long video retrieval, 2022. arXiv:2205.08508. 2

[3] J. Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset, 2017. 2, 4, 5, 6

[4] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600, 2018. 5

[5] Tongjia Chen, Hongshan Yu, Zhengeng Yang, Zechuan Li, Wei Sun, and Chen Chen. Ost: Refining text knowledge with optimal spatio-temporal descriptor for general video recognition, 2024. 2, 5

[6] Yifei Chen, Dapeng Chen, Ruijin Liu, Hao Li, and Wei Peng. Video action recognition with attentive semantic units, 2023. 2, 5, 6

[7] Yifei Chen, Dapeng Chen, Ruijin Liu, Sai Zhou, Wenyuan Xue, and Wei Peng. Align before adapt: Leveraging entity-to-region alignments for generalizable video action recognition, 2024. 2, 5, 6

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. 1

[9] Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen. Clip2video: Mastering video-text retrieval via image clip. *arXiv preprint arXiv:2106.11097*, 2021. 2

[10] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale, 2022. 1

[11] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021. 2

[12] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzyńska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense, 2017. 2, 3

[13] Xiaohu Huang, Hao Zhou, Kun Yao, and Kai Han. Froster: Frozen clip is a strong teacher for open-vocabulary action recognition, 2024. 1, 2, 5

[14] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision, 2021. 1

[15] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding, 2022. 2, 5, 6

[16] Kumara Kahatapitiya, Anurag Arnab, Arsha Nagrani, and Michael S. Ryoo. Victr: Video-conditioned text representations for activity recognition, 2023. 2, 5, 6

[17] Minji Kim, Dongyoon Han, Taekyung Kim, and Bohyung Han. Leveraging temporal contextualization for video action recognition, 2024. 5, 6

[18] Bruno Korbar and Andrew Zisserman. Personalised clip or: how to find your vacation videos. In *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press, 2022. 2

[19] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011. 2

[20] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Limin Wang, and Yu Qiao. Uniformerv2: Spatiotemporal learning by arming image vits with video uniformer, 2022. 1, 2, 6

[21] Kunchang Li, Yali Wang, Yizhuo Li, Yi Wang, Yinan He, Limin Wang, and Yu Qiao. Unmasked teacher: Towards training-efficient video foundation models, 2023. 2

[22] Wei Lin, Leonid Karlinsky, Nina Shvetsova, Horst Possegger, Mateusz Kozinski, Rameswar Panda, Rogerio Feris, Hilde Kuehne, and Horst Bischof. Match, expand and improve: Unsupervised finetuning for zero-shot action recognition with language knowledge. In *ICCV*, 2023. 2, 4, 5

[23] Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard de Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Frozen clip models are efficient video learners. *arXiv preprint arXiv:2208.03550*, 2022. 1, 2, 6

[24] Ruyang Liu, Jingjia Huang, Ge Li, Jiashi Feng, Xinglong Wu, and Thomas H Li. Revisiting temporal modeling for clip-based image-to-video knowledge transferring. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 1, 2

[25] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. CLIP4Clip: An empirical study of clip for end to end video clip retrieval. *arXiv preprint arXiv:2104.08860*, 2021. 1, 2

[26] Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. Expanding language-image pretrained models for general video recognition. 2022. 1, 2, 5, 6

[27] Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. St-adapter: Parameter-efficient image-to-video transfer learning for action recognition. *arXiv preprint arXiv:2206.13559*, 2022. 1, 2, 5, 6

[28] Jungin Park, Jiyoung Lee, and Kwanghoon Sohn. Dual-path adaptation from image to video transformers, 2023. 6

[29] Rui Qian, Shuangrui Ding, and Dahua Lin. Rethinking image-to-video adaptation: An object-centric perspective. 2024. 6

[30] Zhiwu Qing, Shiwei Zhang, Ziyuan Huang, Yingya Zhang, Changxin Gao, Deli Zhao, and Nong Sang. Disentangling

spatial and temporal learning for efficient image-to-video transfer learning. In *ICCV*, 2023. 1, 2, 6

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 1, 2, 5

[32] Hanoona Rasheed, Muhammad Uzair khattak, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Finetuned clip models are efficient video learners. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 1, 2, 3, 4, 5, 6, 7

[33] Yogesh S Rawat Shahzad Ahmad, Sukalpa Chanda. Ez-clip: Efficient zeroshot video action recognition. *arXiv preprint arXiv:2312.08010*, 2024. 5

[34] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012. 2

[35] Shuyuan Tu, Qi Dai, Zuxuan Wu, Zhi-Qi Cheng, Han Hu, and Yu-Gang Jiang. Implicit temporal modeling with learnable alignment for video recognition. In *ICCV*, 2023. 1, 2, 6

[36] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019. 1

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 1

[38] Hao Wang, Fang Liu, Licheng Jiao, Jiahao Wang, Zehua Hao, Shuo Li, Lingling Li, Puhua Chen, and Xu Liu. Vilt-clip: Video and language tuning clip with multimodal prompt learning and scenario-guided optimization, 2024. 5, 6

[39] Luting Wang, Yi Liu, Penghui Du, Zihan Ding, Yue Liao, Qiaosong Qi, Biaolong Chen, and Si Liu. Object-aware distillation pyramid for open-vocabulary object detection, 2023. 2

[40] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *CoRR*, abs/2109.08472, 2021. 1, 2, 5, 6

[41] Mengmeng Wang, Jiazheng Xing, Boyuan Jiang, Jun Chen, Jianbiao Mei, Xingxing Zuo, Guang Dai, Jingdong Wang, and Yong Liu. M2-clip: A multimodal, multi-task adapting framework for video action recognition, 2024. 6

[42] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinyuan Chen, Yaohui Wang, Ping Luo, Ziwei Liu, Yali Wang, Limin Wang, and Yu Qiao. Internvid: A large-scale video-text dataset for multimodal understanding and generation, 2023. 2

[43] Zhaoqing Wang, Yu Lu, Qiang Li, Xunqiang Tao, Yandong Guo, Mingming Gong, and Tongliang Liu. Cris: Clip-driven referring image segmentation, 2022. 2

[44] Zejia Weng, Xitong Yang, Ang Li, Zuxuan Wu, and Yu-Gang Jiang. Open-vclip: Transforming clip to an open-vocabulary video model via interpolated weight optimization. In *ICML*, 2023. 2

[45] Jianzong Wu, Xiangtai Li, Shilin Xu, Haobo Yuan, Henghui Ding, Yibo Yang, Xia Li, Jiangning Zhang, Yunhai Tong, Xudong Jiang, Bernard Ghanem, and Dacheng Tao. Towards open vocabulary learning: A survey. *arXiv pre-print*, 2023. 2

[46] Wenhao Wu, Zhun Sun, and Wanli Ouyang. Revisiting classifier: Transferring vision-language models for video recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2847–2855, 2023. 2

[47] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding, 2021. 2

[48] Taojiannan Yang, Yi Zhu, Yusheng Xie, Aston Zhang, Chen Chen, and Mu Li. AIM: Adapting image models for efficient video action recognition. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 2, 5, 6

[49] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision. *CoRR*, abs/2111.11432, 2021. 1

[50] Renrui Zhang, Rongyao Fang, Peng Gao, Wei Zhang, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021. 2

[51] Wei Zhang, Chaoqun Wan, Tongliang Liu, Xinmei Tian, Xu Shen, and Jieping Ye. Enhanced motion-text alignment for image-to-video transfer learning. In *CVPR*, pages 18504–18515, 2024. 6

[52] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16793–16803, 2022. 2

[53] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*, 2022. 2