What Can Simple Arithmetic Operations Do for Temporal Modeling?

Wenhao Wu[1,2]   Yuxin Song[2]   Zhun Sun[2]   Jingdong Wang[2]   Chang Xu[1]   Wanli Ouyang[3,1]

[1]The University of Sydney   [2]Baidu Inc.   [3]Shanghai AI Laboratory

wenhao.wu@sydney.edu.au   https://github.com/whwu95/ATM

Abstract

Temporal modeling plays a crucial role in understanding video content. To tackle this problem, previous studies built complicated temporal relations through time sequence thanks to the development of computationally powerful devices. In this work, we explore the potential of four simple arithmetic operations for temporal modeling. Specifically, we first capture auxiliary temporal cues by computing addition, subtraction, multiplication, and division between pairs of extracted frame features. Then, we extract corresponding features from these cues to benefit the original temporal-irrespective domain. We term such a simple pipeline as an Arithmetic Temporal Module (ATM), which operates on the stem of a visual backbone with a plug-and-play style. We conduct comprehensive ablation studies on the instantiation of ATMs and demonstrate that this module provides powerful temporal modeling capability at a low computational cost. Moreover, the ATM is compatible with both CNNs- and ViTs-based architectures. Our results show that ATM achieves superior performance over several popular video benchmarks. Specifically, on Something-Something V1, V2 and Kinetics-400, we reach top-1 accuracy of 65.6%, 74.6%, and 89.4% respectively. The code is available at https://github.com/whwu95/ATM.

1. Introduction

Owing to the rapid development of the Internet and mobile devices, researchers can now work with massive amounts of video data. The time-sequenced data provides substantial information in understanding visual tasks such as human activity recognition. While significant progress has been made in extracting the semantics from a single image, how to model the temporal information effectively becomes an essential problem for video recognition.

Historically, temporal information has been constructed either by utilizing low-level streams (*e.g.* optical flow, motion vector) [14, 46, 58, 78], or by building segregated time sequence with recurrent networks using features extracted from an image backbone [8, 25, 76]. After that,

Video frames　　　**Arithmetic Operations on frames**

(a) *t*-th *frame*　　　(c) *Addition*　　　(d) *Subtraction*

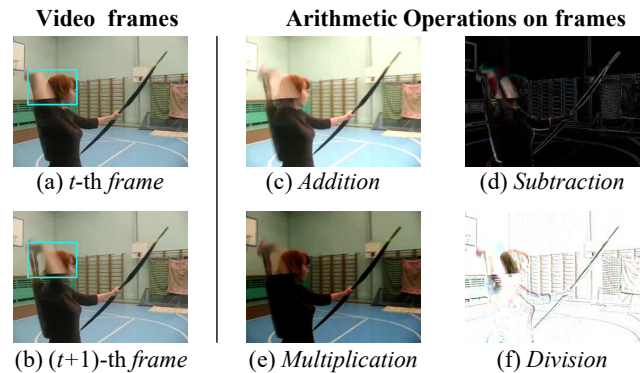(b) (*t*+1)-th *frame*　　(e) *Multiplication*　　(f) *Division*

Figure 1. (a),(b): A pair of consecutive video frames where a cyan rectangle highlights the area around a moving hand. (c)-(f): Visualization of arithmetic operations (✚, ➖, ✖, ➗) on consecutive frames. Zoom in for the best view.

more works present 3D CNNs [5, 51] and factorized 2D+1D CNNs [40, 53, 72] as the natural evolution from their 2D counterparts for dealing with 3D volumetric video data directly, thanks to the rapid improvement of computational capacity. Along this research line, many recent pieces of research follow the "2D backbone + temporal interaction" paradigm to develop efficient temporal modules [17, 23, 24, 27, 28, 29, 32, 34, 50, 57, 61, 68, 83]. The strong point of this approach is that, the pre-trained image backbone can serve as a reasonable initialization, reducing training time and enabling the optimization of the entire video architecture in an end-to-end manner. Under this paradigm, the ways of temporal modeling can be roughly divided into two categories based on the type of interaction between frames: i) sequence-wise modeling (*e.g.*, temporal convolution [13, 29, 60, 61], dynamic convolution [34, 68], and temporal transformer [1, 2, 33]); ii) pair-wise modeling (*e.g.*, pseudo optical flow estimation [50], similarity estimation [23, 24, 56], *etc.*).

We bring the latter into focus. Precisely, pair-wise modeling employs the relationship between an anchor frame and the others. Such relationships are considered to contain expert knowledge. For instance, a few previous works [23, 24, 56] have used correlation operations to estimate the similar-

ity between two frames, while another work [50] has combined subtraction and Sobel operations to estimate optical flow. Despite the effectiveness of these limited attempts on CNN backbones [18], the pair-wise modeling has not received enough attention and exploration when compared to sequence-wise modeling, especially in the current era of the rising popularity of vision transformer backbones [9, 31]. In this study, we aim to fill this gap by revisiting the pair-wise relationship with the most basic arithmetic operations: *Addition* (✚), *Subtraction* (━), *Multiplication* (✖), and *Division* (➗), without extensive elaborate design.

The arithmetic operations are commonly employed in ordinary vision and time series tasks, and their outputs have explainable physical meaning. Briefly, addition is used to implement integral to obtain properties such as the average intensity of an image [7] or the accumulated status of a feature sequence; On the contrary, subtraction detects changes over time and approximates the tendency of the motion such as optical flow [50]; Next, multiplication can be considered as a measure of similarity or correlation between (regions of) vectorized features of frames [22], such that the features that are unchanged along time shall be highlighted; Finally, on the opposite of multiplication, the division between frames locates the features that are changed intensely [55]. For easier understanding, we provide an intuition of the physical meaning of these operations using the original frames, as depicted in Figure 1. Notably, our work focuses on exploring these arithmetic operations on *encoded feature representations*, rather than on raw frames.

In this work, we raise a question: *Can simple arithmetic operations be employed for temporal modeling?* To answer this question, we first abstract a simple video framework that divides the neural network into a temporal-irrespective stem and a temporal-interactive block. This enables the implementation of the temporal-irrespective stem using existing vision backbones, such as CNNs or Vision Transformers. We then propose an **A**rithmetic **T**emporal **M**odule (ATM), which conducts arithmetic operations between pairs of frame features to generate auxiliary temporal cues and then transform them back onto the network stem. We organize comprehensive ablation studies on the instantiation of ATMs, *w.r.t* the choice of operations, temporal context range, structures of feature extractor, and the attached locations. Subsequently, we discover the composite of *Subtraction* and *Multiplication* achieve the best performance among the candidates. Finally, we evaluate the proposed method by instantiating it on both representative CNN and Transformer backbones (*i.e.*, ResNet [18], ViT [9]), to show its generality and superiority. Our contributions are as follows:

- We explore the ascendency of employing fundamental arithmetic operations for temporal modeling, and propose an ad-hoc block named **A**rithmetic **T**emporal **M**odule (ATM) that integrates pair-wise temporal in-

teractions between pairs of frame features.

- We conduct a comprehensive exploration of the instantiation of ATMs on both CNN backbones and Vision Transformer backbones.

- We demonstrate that existing backbones plugged with ATM can achieve superior performance on a broad range of popular benchmarks. Specifically, our method achieves Top-1 accuracy of 65.6% on Something-Something V1, 74.6% on Something-Something V2, and 89.4% on Kinetics-400, respectively.

2. Related Work

CNNs for Video Recognition. Convolutional neural networks (CNNs) have made significant contributions to the field of video recognition. Initially, temporal information was constructed using low-level streams (*e.g.*, optical flow, motion vector, RGB difference) [14, 46, 58, 78], or through the use of segregated time sequences with recurrent networks that extracted features from an image backbone [8, 25, 76]. As computational capacity rapidly improved, 3D CNNs [5, 51] and factorized 2D+1D CNNs [13, 40, 53, 72] emerged as natural extensions of their 2D counterparts, allowing for the direct handling of 3D volumetric video data. To further alleviate the computational cost of video classification networks, many studies [23, 28, 29, 32, 34, 35, 47, 57, 61] have aimed to design efficient temporal modules and incorporate them into 2D CNNs [18, 19]. There are also studies have developed efficient video architectures that do not rely on existing 2D CNNs, resulting in even more lightweight structures [12, 52, 56]. Additionally, a few works [68, 80] have introduced 4D convolutional techniques to capture relationships between clips. Another research direction is orthogonal to the aforementioned CNN-based video architecture, which focuses on dynamic inference [62, 63, 69, 70, 71, 81] to achieve efficient recognition.

Among the aforementioned studies, [23, 24, 56] employ correlation to compute the similarity between consecutive frames and incorporate it into the design of CNN-based temporal modeling. We consider these works to be relevant to ours since we view correlation as a specific instance of multiplication. However, apart from differences in implementation details, our goal is to explore the potential of various cues generated from the simple arithmetic operations and propose a universal temporal module applicable to both CNNs and Vision Transformers, rather than designing customized CNN-based video architectures.

Transformers for Video Recognition. Transformers [54], originally proposed for the language domain, have been adapted for vision tasks and have gained popularity in the computer vision community. Image Transformers [9, 16, 31] have achieved great success, leading to the investigation of transformers for video recognition. Several works
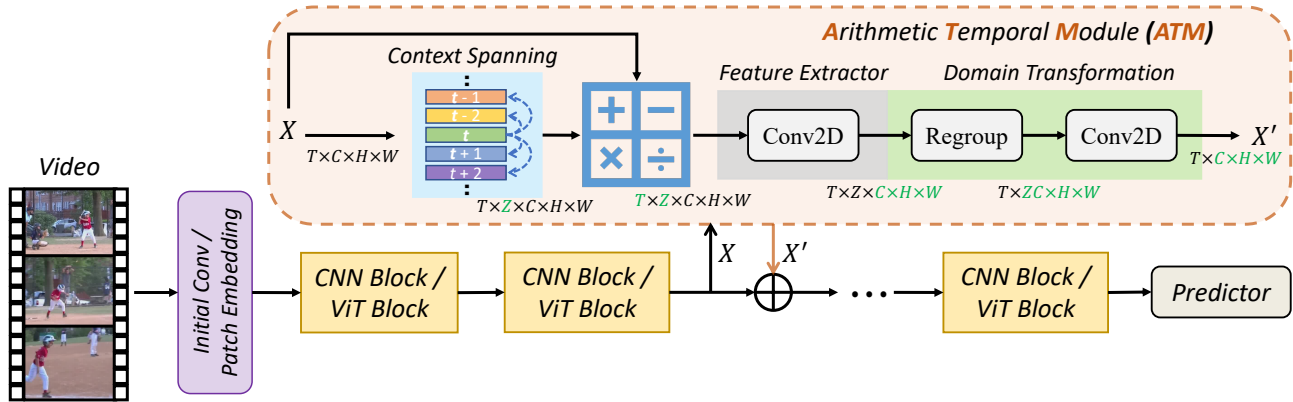
Figure 2. The overall framework of the simple **A**rithmetic **T**emporal **M**odule (ATM) for temporal modeling. The key motivation behind ATM is to explore the potential of simple arithmetic operations to capture auxiliary temporal clues that may be embedded in current video features, without relying on the elaborate design. The ATM can be integrated into both vanilla CNN backbone (*e.g.*, ResNet [18]) and Vision Transformer (*e.g.*, ViT [9]) for video action recognition.

[1, 2, 10, 33, 37] propose different spatial and temporal attention designs using the large-scale ImageNet-21K [42] pre-trained model or designing new models from scratch, demonstrating improved performance for spatiotemporal modeling for video recognition.

In addition to the evolution of video transformer architectures, large-scale pre-training has significantly boosted their popularity. Several studies [1, 43, 73, 79] have achieved remarkable performance in video recognition task using the JFT-300M [48] or even JFT-3B [77] pre-trained image transformer. More recently, large-scale image-text pre-training has attracted wider attention. One representative work is CLIP [41], which leverages 400M image-text pairs (namely, WIT-400M) to train vision and text encoders using contrastive learning. This breakthrough has inspired several research endeavors that directly explore the application of CLIP models in text-video retrieval [11, 36, 64, 82], while others have also attempted to utilize CLIP for video recognition. Based on whether CLIP's text encoder is utilized, these studies can be categorized into two lines: the video-text paradigm [20, 38, 59, 65, 66, 67] and the video-only paradigm [30, 39, 74]. In our research, we only use CLIP's vision encoder without involving any textual knowledge for additional gains. Various CLIP variants have subsequently emerged. For example, Florence [75] utilizes 900M image-text pairs, while EVA-CLIP [49] surpasses that by leveraging an impressive 2B image-text pairs (namely, Merged-2B) for pre-training.

3. Approach

In this section, we begin by providing an overview of the **A**rithmetic **T**emporal **M**odule (ATM), followed by the introduction of its structure. We then provide a detailed explanation of the arithmetic operations employed in ATM.

3.1. Overview of ATM

The key motivation behind ATM is to leverage the potential of simple arithmetic operations that do not require parameters to capture additional temporal information present in current frame features. This approach provides a more straightforward and interpretable way to capture additional temporal clues without the need for elaborate design. By integrating ATM into various backbones, including the popular CNN - ResNet [18] and Vision Transformer - ViT [9], effective temporal modeling can be achieved.

Specifically, ATM consists of several stages. Firstly, context spanning is performed on temporal-irrespective frame features to construct context frames. Subsequently, arithmetic operations (**+**, **−**, **×**, **÷**) are applied to the features of each frame and its context frames to obtain temporal clues. These temporal clues are then fed into a feature extractor to extract their features. Finally, the temporal clues are projected back to the original temporal-irrespective domain through domain transformation, and are connected to the original features in a residual form. The overall framework is illustrated in Figure 2.

3.2. Arithmetic Temporal Module (ATM)

The Structure of ATM. Here we describe the specific structure of the ATM. Given a video, we sample T frames for video recognition. To capture the temporal information, we propose to explicitly utilize the relationship between context frames to capture the temporal information. Specifically, we use *Context Spanning* to construct Z context frames for each frame, where Z defines the context range. For clarity, we define the pair-wise *Arithmetic Operation* (any of **+**, **−**, **×**, **÷**) as a function $\psi(\cdot, \cdot)$ that performs point-to-point operations on two frame features in the spatial dimensions H and W. The output of the arithmetic

operation is formulated as follows:

$$X^{\text{OUT}} = \texttt{Concatenate}\Big([\psi(X_t, X_z), \dots]\Big),$$
$$t = 1, 2 \dots T, z \in \mathcal{Z} \quad (1)$$

where $X_t \in \mathbb{R}^{C \times H \times W}$ represents the feature of the anchor frame, while $X_z \in \mathbb{R}^{C \times H \times W}$ represents the feature of one of its spanned context frame. $C, H,$ and W denote the number of channels, height, and width of a channel slice, respectively. $\mathcal{Z}$ denotes the index set of context frames. If we define Z as the number of context frames in $\mathcal{Z}$, the output feature X^{OUT} becomes a tensor of size $\mathbb{R}^{T \times Z \times C \times H \times W}$. The context frames are chosen based on context range Z to interact with the anchor frame:

- When $Z = 1$, the context is chosen as the next frame, $\mathcal{Z} = \{t + 1\}$.

- When $Z = 2, 4, 6 \dots$, the context is chosen as the previous $\frac{Z}{2}$ and next $\frac{Z}{2}$ frames, *e.g.*, $\mathcal{Z} = \{t - 1, t + 1\}$ for $Z = 2$.

After performing the arithmetic operation for temporal interaction, we obtain a composite temporal signal of size $\mathbb{R}^{T \times Z \times C \times H \times W}$, where the $T \times Z$ matrix records the pairwise temporal cues for each frame. We then employ a ***Feature Extractor*** that extracts spatially dependent representations while preserving the $T \times Z$ matrix. In practice, we implement the feature extractor using a stack of two 3×3 convolutions over the spatial axes.

Finally, we perform ***Domain Transformation*** to project the representations back to the temporal-irrespective domain. Specifically, we regroup the pair-wise interaction axis Z with the spatial channel C. To reduce the dimensionality of the representations from ZC to C, we employ a 1×1 convolution over the channel dimension. The resulting features are then connected to the original features $X \in \mathbb{R}^{T \times C \times H \times W}$ in a residual form, providing effective temporal information to the original features.

Integrating ATM into CNNs and Vision Transformers. To showcase the versatility of our simple ATM, we integrate it into existing image backbones, enabling powerful temporal modeling. We specifically select two widely-used backbone architectures: CNNs and Vision Transformers. In our experiments, we incorporate ATM into vanilla ResNet [18] and ViT [9], which are representative architectures in their respective domains. For ResNet, we insert one ATM after the last residual block of res$_3$. For ViT, we integrate one ATM after the 7-th layer of ViT-B (which has 12 layers) and after the 14-th layer of ViT-L (which has 24 layers).

3.3. Arithmetic Operations

In Section 3.2, we present the structure of our ATM. Now, let's delve into the instantiation of the arithmetic operations employed within ATM. Specifically, we consider four fundamental arithmetic operations: $\boldsymbol{+}, \boldsymbol{-}, \boldsymbol{\times}, \boldsymbol{\div}$.

(1) *Addition* ($\boldsymbol{+}$): We use the addition operation to compute the accumulated signals of actions. Intuitively, it captures the spatial occupancy of feature representations, *i.e.*, patterns of a series of actions, which is helpful in understanding actions like gestures. The addition of A with B at spatial location (c, h, w) is computed as

$$Addition(A, B)_{(c,h,w)} = A(c, h, w) + B(c, h, w), \quad (2)$$

yielding an *Addition* tensor of size $C \times H \times W$.

(2) *Subtraction* ($\boldsymbol{-}$): The subtraction operation is widely used to estimate optical flow, where it represents the magnitude of the spatial feature change. Here, we use the subtraction operation to infer the smoothness of the change by concatenating the subtraction, which helps us understand the action dynamics. The subtraction of A with B at spatial location (c, h, w) is computed as

$$Subtraction(A, B)_{(c,h,w)} = A(c, h, w) - B(c, h, w), \quad (3)$$

yielding a *Subtraction* tensor of size $C \times H \times W$.

(3) *Multiplication* ($\boldsymbol{\times}$): The spatial multiplication between two features can be considered as the apparent similarity of spatial features. The similarity signals capture the likelihood of the spatial appearance of a feature in the interval of the paired frames, and can be useful in understanding the dynamics of actions. In this paper, we utilize the local multiplication of a $P \times P$ neighborhood instead of global multiplication. This operation is also referred to as correlation in some cases. Concretely, we compute the multiplication of A with B at spatial location (h, w) as

$$Multiplication(A, B)_{(h,w)} = A(h, w) \cdot B(h + i, w + j),$$
$$(i, j) \in P \times P. \quad (4)$$

Here, we use $P \times P$ to denote the relative spatial location set of neighborhoods. For $P = 2k + 1$, $\{P \times P\} \equiv \{-k, \dots, k\} \times \{-k, \dots, k\}$, where k represents the maximum offset within the neighborhood. We set P to 9 in this paper empirically. Thus, we obtain a *Multiplication* tensor of size $P^2 \times H \times W$, which is transformed to the size of the original feature $C \times H \times W$ via several convolutional layers.

(4) *Division* ($\boldsymbol{\div}$): The element-wise division operation amplifies the changes in features between a frame and its anchor frame. This operation functions similarly to subtraction, but the magnitude of the difference is standardized, allowing even features with small numerical values to result in a large quotient. To stabilize the training process, we employ the subtraction between the $\texttt{log}$ value of the features, which can be expressed as

$$Division(A, B)_{(h,w)} = log(A(c, h, w) + \epsilon) -$$
$$log(B(c, h, w) + \epsilon), \quad (5)$$

where ϵ is a stability parameter set to 1 in our experiments, which avoids a non-positive value in the logarithm operator.

4. Experiments

4.1. Datasets and Evaluation Metrics

We evaluate our ATM on five video recognition benchmarks, including Something-Something (SS) V1&V2 [15], Kinetics-400 (K400) [21], ActivityNet [4], and Charades [45]. The SS V1 contains 110k videos, while V2 contains 220k video clips across 174 fine-grained classes. In contrast, Kinetics-400 is another widely-used large-scale video dataset, consisting of 300k video clips across 400 human action categories. In *Appendix*, we also demonstrate the generalization ability of our ATM by evaluating it on smaller datasets, *e.g.*, ActivityNet and Charades.

Our results are reported according to the official evaluation metrics. We report top-1 and top-5 accuracy for K400 and SS V1&V2, respectively. For ActivityNet and Charades, we report the mean average precision (mAP). In addition, we report the computational cost (in FLOPs) to provide a comprehensive understanding of model complexity. Note that we only employ RGB frames for all datasets.

4.2. Implementation Details

Training. In this paper, we employ ResNet [18] and ViT [9] as representative backbones for CNNs and Vision Transformers, respectively. For CNN-based instantiation, we utilize the widely-used ImageNet-1K [6] pre-trained ResNet, which aligns with common practices in prior studies [29, 61, 57]. In the case of Transformer-based instantiation, we follow the recent trend [30, 39, 74] by adopting the visual encoder of CLIP [41] as the backbone. Our models are trained using either 8 or 16 frames, and we provide detailed training configurations in *Appendix*.

Inference. To trade off accuracy and speed, we consider two evaluation protocols. (1) *Single View*: We use only 1 clip per video and the center crop for efficient evaluation. (2) *Multiple Views*: It is a common practice [5, 13, 61] to sample multiple clips per video with several spatial crops to get higher accuracy. For readers' reference, we present our results with various views in the *Appendix*.

4.3. Ablation Studies

We conduct ablation studies on both ResNet and ViT to evaluate the temporal modeling capability of our ATM. The Something-Something dataset involves complex object-object and human-object interactions, which require capturing strong temporal relationships and motion cues for accurate recognition. Therefore, recognition based solely on scene information is challenging for this dataset. The results of our studies are presented in Table 1 and Table 2.

Four Arithmetic Operations. We investigate the impact of four arithmetic operations - Addition (✚), Subtraction (━), Multiplication (✖), and Division (➗) - on the performance of our proposed method for temporal modeling. We also compare the results of each operation with a baseline approach that uses mean pooling to model the temporal information of videos, focusing only on appearance features. The results shown in Tables 1a and 2a demonstrate that each arithmetic operation significantly outperforms the baseline, indicating that simple arithmetic operations can provide strong temporal cues for effective temporal modeling. We also observe that ✖, ━, and ➗ lead to relatively better results compared to ✚. These results are consistent across both ResNet and ViT backbones.

Fusion of Multiple Temporal Cues. We investigate the redundancy and complementarity of various sources of temporal information by stacking different instances of ATM. Since ✖ achieves the highest performance, we primarily adopt it and combine it with other operations. Based on Tables 1a and 1b, we observe that all three other operations complement ✖ to a certain extent, with ━ and ✖ showing the most prominent complementary effect. We also find that ✚ and ━ can complement each other. Therefore, we combine ✚, ━, and ✖, but do not observe any additional improvement. Similarly, as shown in Tables 2a and 2b, the fusion of ━ and ✖ on ViT improves their individual results.

Temporal Context Range. We investigate the impact of varying the temporal context range Z on pair-wise operations and present the results in Table 1c. The context range determines the scope of obtaining temporal cues for each frame from its adjacent frames. For instance, setting $Z = 4$ implies that each frame undergoes arithmetic operations with its two preceding and two succeeding frames. We observe that a larger context range can enhance the non-context model, resulting in noticeable performance gain. However, increasing Z beyond 4 did not yield a significant improvement. Hence, we set Z to 4 as the default value.

Complementarity with Temporal Convolution. Temporal convolution (T-Conv) is a common and efficient temporal operation used in previous methods [28, 29, 32, 61]. It performs a $3 \times 1 \times 1$ depth-wise convolution along the temporal dimension. To incorporate T-Conv into our models, we insert T1D into each block of ResNet's res4 and res5, and apply T1D to the [CLS] token in all layers of ViT, which is a commonly used practice. We investigate the complementarity between ATM and T-Conv. Table 1d and 2b show that although T-Conv performs worse than ATM on both ResNet and ViT, combining ATM with T-Conv further improves performance. Therefore, we include the lightweight T-Conv in our default setting for subsequent experiments.

Feature Extractor. We also investigate the impact of different feature extractors in our framework. We experiment with three common instantiations: Fully Connected layer

+	−	×	÷	Top-1
-	-	-	-	16.5%
✓	-	-	-	41.1%
-	✓	-	-	43.6%
-	-	✓	-	**46.5%**
-	-	-	✓	43.1%

(a) Study on the effects of four arithmetic operations independently.

+	−	×	÷	Top-1
✓	-	✓	-	47.8%
-	✓	✓	-	**48.2%**
-	-	✓	✓	47.3%
✓	✓	-	-	44.6%
✓	✓	✓	-	48.1%

(b) Study on the complementarity among four temporal interactions.

Z	Top-1	FLOPs
0	16.5%	14.5G
1	43.7%	15.4G
2	47.0%	16.5G
4	**48.2%**	17.7G
6	48.0%	18.8G

(c) Study on the temporal context range Z.

Method	Top-1
T-Conv	44.3%
ATM (−, ×)	48.2%
ATM + T-Conv	**48.7%**

(d) Exploring compatibility of ATM with Temporal Convolution (T-Conv).

	Top-1
FC	47.8%
MLP	48.1%
Conv Stack	**48.7%**

(e) Study on various feature extractor in ATM. MLP: FC×3. Conv Stack: Conv2D×2.

Stage	Top-1	FLOPs
res_2	48.8%	26.5G
res_3	**48.7%**	17.7G
res_4	47.6%	17.9G
res_5	43.1%	15.5G

(f) Study on the location of ATM.

Model	#Frame	Backbone	Top-1	Top-5
TSM [29]	8	ResNet-50	47.1%	76.2%
Ours	8	ResNet-18	48.7%	77.7%
	8	ResNet-50	53.9%	81.8%
	16	ResNet-50	56.3%	83.4%

(g) Study on deeper ResNet backbones and more frames.

Table 1. Ablation studies with the representative **CNN backbone - ResNet** [18] on Something-Something V1. Unless otherwise specified, all models use ResNet18 with 8 frames under the single view protocol.

+	−	×	÷	Top-1
-	-	-	-	27.3%
✓	-	-	-	51.3%
-	✓	-	-	53.8%
-	-	✓	-	**54.5%**
-	-	-	✓	53.7%

(a) Study on the effects of four arithmetic operations independently.

Method	Top-1
ATM (−, ×)	56.1%
T-Conv	51.3%
ATM + T-Conv	**58.1%**

(b) Exploring compatibility of ATM with Temporal Convolution (T-Conv).

Position	Top-1
layer{2}	36.9%
layer{4}	55.7%
layer{6}	57.5%
layer{7}	**58.1%**
layer{8}	57.1%
layer{10}	55.7%

(c) The location of ATM.

#Frame	Backbone	Top-1
8	ViT-B	58.1%
16	ViT-B	59.5%
32	ViT-B	60.9%
8	ViT-L	61.9%
16	ViT-L	63.3%

(d) Study on larger ViT backbones and more frames.

Table 2. Ablation studies with representative **Vision Transformer - ViT** [9] on Something-Something V1. Unless otherwise specified, all models use ViT-B/16 with 8 frames under the single view protocol.

(FC), MultiLayer Perceptron (MLP), and a stack of Convolutional layers (Conv Stack). As shown in Table 1e, the Conv Stack achieves the highest accuracy of 48.7%, outperforming the other two options with 47.8% and 48.1% for FC and MLP, respectively. Therefore, we select the Conv Stack as the default feature extractor in our experiments.

The Optimal Position for ATM. We investigate the optimal position for inserting the ATM in ResNet and ViT. We denote the conv2_x to conv5_x layers of ResNet architecture as res_2 to res_5. To gradually incorporate the ATM into ResNet-18, we add it from res_2 to res_5. For example, res_2 denotes that one ATM is inserted after the last res block of res_2. As shown in Table 1f, adding ATM to res_2 or res_3 results in significant improvements. Performance is minimally impacted when ATM is added to res_2 or res_3, and thus we use ATM in res_3 for efficiency. Regarding ViT, we also observe that inserting ATM into the middle layers leads to better performance, as shown in Table 2c. Thus, we insert one ATM after the 7-th layer of ViT-B (which has 12 layers) and after the 14-th layer of ViT-L (which has 24 layers).

More Instantiations. We evaluate our method using various visual encoders and more input frames. Tables 1g and 2d present the results of different instantiations based on ResNet and ViT, respectively. Notably, our method with ResNet-18 outperforms TSM-ResNet50 [29] (48.7% *vs.* 47.1%). In general, we observe that deeper backbones can achieve better performance, and increasing the number of frames can lead to additional improvements.

4.4. Main Results

Until 2021, ResNet was the primary backbone for video recognition methods. However, since 2022, the landscape has shifted with the advent of new techniques utilizing vision transformers that incorporate large-scale pre-training [41, 42, 48, 49, 77]. In this section, we present the results of our method on both ResNet and ViT backbones, and compare them with previous works.

Results on Something-Something V1 & V2. We summarize our results in Table 3. To ensure fair comparisons with CNN-based methods, we evaluate our ATM using differ-

Methods	Year	Backbones	Pre-training	Frames×Crops×Clips	GFLOPs	SSV1	SSV2
TSM [29]	ICCV'2019	ResNet50	ImageNet-1K	16×1×1	65	47.2%	63.4%
TSM$_{En}$ [29]	ICCV'2019	ResNet50	ImageNet-1K	(8+16)×1×1	98	49.7%	N/A
TEINet$_{En}$ [32]	AAAI'2020	ResNet50	ImageNet-1K	(8+16)×1×1	99	52.5%	66.5%
TEA [28]	CVPR'2020	ResNet50	ImageNet-1K	16×3×10	70×30	52.3%	N/A
MSNet [23]	ECCV'2020	ResNet50	ImageNet-1K	16×1×1	67	52.1%	64.7%
MVFNet$_{En}$ [61]	AAAI'2021	ResNet50	ImageNet-1K	(8+16)×3×2	99×6	54.0%	66.3%
TDN$_{En}$ [57]	CVPR'2021	ResNet50	ImageNet-1K	(8+16)×1×1	108	55.1%	67.0%
SELFY$_{En}$ [24]	ICCV'2021	ResNet50	ImageNet-1K	(8+16)×1×1	114	55.8%	67.4%
Ours		ResNet50	ImageNet-1K	16×1×1	74	56.3%	67.4%
Ours	ICCV'2023	ResNet50	ImageNet-1K	32×1×1	148	57.1%	68.4%
Ours$_{En}$		ResNet50	ImageNet-1K	(8+16)×1×1	111	57.6%	68.3%
CorrNet [56]	CVPR'2020	ResNet101	ImageNet-1K	32×3×10	224×30	51.7%	N/A
TDN [57]	CVPR'2021	ResNet101	ImageNet-1K	16×1×1	132	55.3%	66.9%
TDN$_{En}$ [57]	CVPR'2021	ResNet101	ImageNet-1K	(8+16)×1×1	198	56.8%	68.2%
Ours		ResNet101	ImageNet-1K	16×1×1	134	57.2%	68.2%
Ours$_{En}$	ICCV'2023	ResNet101	ImageNet-1K	(8+16)×1×1	201	58.6%	69.4%
Ours$_{En}$		ResNet101	ImageNet-1K	(8+16+32)×1×1	469	**60.0%**	**70.8%**
ViViT [1]	ICCV'2021	L/16×2 FE	IN-21K	32×3×14	47600	N/A	65.4%
MTV [73]	CVPR'2022	MTV-B (320↑)	IN-21K	32×3×4	11200	N/A	68.5%
CoVeR [79]	ArXiv'2022	TSFormer (448↑)	JFT-3B	16×3×1	17600	N/A	70.8%
EVL [30]	ECCV'2022	ViT-B/16	WIT-400M	32×3×1	2047	N/A	62.4%
EVL [30]	ECCV'2022	ViT-L/14	WIT-400M	32×3×1	9641	N/A	66.7%
AIM [74]	ICLR'2023	ViT-B/16	WIT-400M	32×3×1	2496	N/A	69.1%
AIM [74]	ICLR'2023	ViT-L/14	WIT-400M	32×3×1	11508	N/A	70.6%
ST-Adapter [39]	NeurIPS'2022	ViT-L/14	WIT-400M	32×3×1	8248	N/A	72.3%
UniFormerV2 [26]	ICCV'2023	ViT-L/14	WIT-400M	32×3×1	5200	62.7%	73.0%
Ours		ViT-B/16	WIT-400M	32×3×1	378×3	61.3%	71.9%
Ours	ICCV'2023	ViT-L/14	WIT-400M	16×3×1	842×3	63.7%	73.2%
Ours		ViT-L/14	WIT-400M	16×3×2	842×6	64.0%	73.5%
Ours		ViT-L/14	Merged-2B	16×3×2	842×6	**65.6%**	**74.6%**

Table 3. Comparison with the SOTAs on Something-Something (SS) V1 & V2. Unless otherwise specified, all models use a spatial size of 224×224 as the input. Top-1 accuracy is reported. En denotes the ensemble of multiple models with different numbers of input frames.

ent configurations, including various backbones (ResNet-50/101) and numbers of input frames (8/16). Our ATM consistently outperforms previous state-of-the-art works presented at top-tier conferences on both datasets. For instance, on SS V1, we achieve better results than TDN [57] using both ResNet-50 (**57.6%** *vs.* 55.1%) and ResNet-101 (**58.6%** *vs.* 56.8%) backbones. Similar conclusions can be drawn for SS V2. We also compare our method with recent works that utilize large-scale pre-trained video transformers [30, 74, 39], which use the same CLIP pre-trained backbone. Our approach shows significant improvements in accuracy, with an absolute improvement of 9% (**71.9%** *vs.* 62.4%), while utilizing fewer FLOPs (1188G *vs.* 2047G) compared to EVL [30] with the same number of frames. In comparison to more recent methods such as AIM [74], ST-Adapter [39] and UniFormerV2 [26], our approach consis-

tently outperforms them. Moreover, when we significantly increase the scale of the pre-training data from 400 million to 2 billion samples (namely Merged-2B [49], which merges 1.6 billion samples from the LAION-2B [44] dataset with 0.4 billion samples from the COYO-700M [3] dataset), our method exhibits substantial improvement. Remarkably, our method outperforms CoVeR [79] with a significant margin (**74.6%** *vs.* 70.8%), despite their adoption of a larger spatial resolution (448 *vs.* 224), a pre-training dataset scale that is 1.5 times larger (3B *vs.* 2B), and computational costs that are 7 times greater (17.6T *vs.* 2.5T).

Results on Kinetics-400. To verify the generalization ability of our method, we conduct experiments on the Kinetics-400 dataset, which does not rely as heavily on temporal modeling as the SS dataset. Table 4 presents our comparison with both CNN-based and transformer-based

Methods	Year	Backbones	Pre-training	Frames×Crops×Clips	GFLOPs	Top-1	Top-5
TSM [29]	ICCV'2019	ResNet-50	ImageNet-1K	8×3×10	33×30	74.1%	91.2%
TEINet [32]	AAAI'2020	ResNet-50	ImageNet-1K	8×3×10	33×30	74.9%	91.8%
TEA [28]	CVPR'2020	ResNet-50	ImageNet-1K	8×3×10	33×30	75.0%	91.8%
MVFNet [61]	AAAI'2021	ResNet-50	ImageNet-1K	8×3×10	32.9×30	76.0%	92.4%
TANet [34]	ICCV'2021	ResNet-50	ImageNet-1K	8×3×10	43×30	76.3%	92.6%
TDN [57]	CVPR'2021	ResNet-50	ImageNet-1K	8×3×10	36×30	76.6%	92.8%
Ours	ICCV'2023	ResNet-50	ImageNet-1K	8×3×10	37×30	77.0%	92.9%
Ours		ResNet-50	ImageNet-1K	16×3×10	74×30	**77.6%**	**93.2%**
NL+I3D [60]	CVPR'2018	ResNet-101	ImageNet-1K	128×3×10	359×30	77.7%	93.3%
SmallBig [27]	CVPR'2020	ResNet-101	ImageNet-1K	32×3×4	418×12	77.4%	93.3%
MVFNet [61]	AAAI'2021	ResNet-101	ImageNet-1K	16×3×10	125×30	78.4%	93.4%
TDN [57]	CVPR'2021	ResNet-101	ImageNet-1K	16×3×10	132×30	78.5%	93.9%
Ours	ICCV'2023	ResNet-101	ImageNet-1K	16×3×10	134×30	78.8%	93.7%
Ours		ResNet-152	ImageNet-1K	16×3×10	191×30	79.4%	93.7%
Ours$_{En}$		R101+R152	ImageNet-1K	(16+16)×3×10	326×30	**80.5%**	**94.4%**
ViViT [1]	ICCV'2021	H/16×2	JFT-300M	32×4×3	8316×12	84.8%	95.8%
MTV [73]	CVPR'2022	MTV-H	JFT-300M	32×4×3	3706×12	85.8%	96.6%
Florence [75]	ArXiv'2021	Florence (384↑)	FLD-900M	32×4×3	N/A	86.5%	97.3%
ST-Adapter [39]	NeurIPS'2022	ViT-L/14	WIT-400M	32×1×3	8248	87.2%	97.6%
AIM [39]	ICLR'2023	ViT-L/14	WIT-400M	32×1×3	11208	87.5%	97.7%
EVL [30]	ECCV'2022	ViT-L/14	WIT-400M	32×1×3	8088	87.3%	-
EVL [30]	ECCV'2022	ViT-L/14 (336↑)	WIT-400M	32×1×3	18196	87.7%	-
X-CLIP [38]	ECCV'2022	ViT-L/14 (336↑) [†]	WIT-400M	16×4×3	3086×12	87.7%	97.4%
Text4Vis [66]	IJCV'2023	ViT-L/14 (336↑) [†]	WIT-400M	32×4×3	3829×12	88.4%	98.0%
BIKE [67]	CVPR'2023	ViT-L/14 (336↑) [†]	WIT-400M	32×4×3	3728×12	88.6%	98.3%
Ours		ViT-L/14	WIT-400M	8×3×4	421×12	87.3%	97.4%
Ours	ICCV'2023	ViT-L/14	WIT-400M	32×3×4	1684×12	88.0%	97.6%
Ours		ViT-L/14 (336↑)	WIT-400M	32×3×4	3784×12	88.2%	97.9%
Ours		ViT-L/14 (336↑)	Merged-2B	32×3×4	3784×12	**89.4%**	**98.3%**

Table 4. Comparison with the SOTAs on Kinetics-400. Unless otherwise specified, all models use a spatial size of 224×224 as the input. "IN" indicates ImageNet. [†] denotes that the method utilizes textual knowledge from the text encoder of CLIP [41].

methods. Our ATM consistently outperforms CNN-based methods using both ResNet-50 and ResNet-101 backbones, showcasing its effectiveness in capturing temporal dynamics. Moreover, we conducted a comprehensive comparison with recent transformer-based methods that utilize web-scale pre-training. Notably, utilizing the same CLIP pre-trained backbones, our ATM achieved a substantial performance improvement over EVL [30], AIM [74], and ST-Adapter [39], even when using fewer frames. It is worth mentioning that our method even performs comparably with more recent state-of-the-art approaches such as Text4Vis [65, 66] and BIKE [67], which integrate additional textual knowledge. Additionally, our method achieves superior performance compared to methods pre-trained with JFT-300M [48] or FLD-900M [75], while requiring less computational cost. Furthermore, with the significant scale-up of the pre-training data to 2 billion samples, our method attains an outstanding top accuracy of 89.4%, solidifying its

position as a state-of-the-art approach.

5. Conclusions

This paper provides an answer to the title question: The arithmetic operations between pairs of frame features are useful in temporal modeling by capturing expert knowledge. To achieve this, we propose an Arithmetic Temporal Module (ATM) that performs addition, subtraction, multiplication, and division between pairs of extracted frame features, generating temporal cues. Then, we extract corresponding features from these cues back to the temporal-irrespective backbone. We comprehensively explore the instantiation of ATM on both CNN and Vision Transformer backbones and find that the combination of subtraction and multiplication yields the best performance on video recognition tasks. Finally, we demonstrate that existing backbones plugged with ATM can achieve superior performance on a broad range of popular video benchmarks.

Acknowledgments

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. *ICCV*, 2021. 1, 3, 7, 8

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, pages 813–824. PMLR, 2021. 1, 3

[3] Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. Coyo-700m: Image-text pair dataset. https://github.com/kakaobrain/coyo-dataset, 2022. 7

[4] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 5

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 1, 2, 5

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[7] Konstantinos G Derpanis. Integral image-based representations. *Department of Computer Science and Engineering York University Paper*, 1(2):1–6, 2007. 2

[8] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015. 1, 2

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 3, 4, 5, 6

[10] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *ICCV*, 2021. 3

[11] Bo Fang, Wenhao Wu, Chang Liu, Yu Zhou, Yuxin Song, Weiping Wang, Xiangbo Shu, Xiangyang Ji, and Jingdong Wang. Uatvr: Uncertainty-adaptive text-video retrieval. In *ICCV*, 2023. 3

[12] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *CVPR*, pages 203–213, 2020. 2

[13] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. *ICCV*, 2019. 1, 2, 5

[14] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Spatiotemporal residual networks for video action recognition. In *NeurIPS*, 2016. 1, 2

[15] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *ICCV*, 2017. 5

[16] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. In *NeurIPS*, volume 34, 2021. 2

[17] Dongliang He, Zhichao Zhou, Chuang Gan, Fu Li, Xiao Liu, Yandong Li, Limin Wang, and Shilei Wen. Stnet: Local and global spatial-temporal modeling for action recognition. In *AAAI*, 2019. 1

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 3, 4, 5, 6

[19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 2

[20] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *ECCV*, 2022. 3

[21] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 5

[22] BVK Vijaya Kumar, Abhijit Mahalanobis, and Richard D Juday. *Correlation pattern recognition*. Cambridge university press, 2005. 2

[23] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. Motionsqueeze: Neural motion feature learning for video understanding. In *ECCV*, pages 345–362. Springer, 2020. 1, 2, 7

[24] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. Learning self-similarity in space and time as generalized motion for video action recognition. In *ICCV*, pages 13065–13075, 2021. 1, 2, 7

[25] Fu Li, Chuang Gan, Xiao Liu, Yunlong Bian, Xiang Long, Yandong Li, Zhichao Li, Jie Zhou, and Shilei Wen. Temporal modeling approaches for large-scale youtube-8m video understanding. *arXiv preprint arXiv:1707.04555*, 2017. 1, 2

[26] 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. In *ICCV*, 2023. 7

[27] Xianhang Li, Yali Wang, Zhipeng Zhou, and Yu Qiao. Smallbignet: Integrating core and contextual views for video classification. In *CVPR*, pages 1092–1101, 2020. 1, 8

[28] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for action recognition. In *CVPR*, pages 909–918, 2020. 1, 2, 5, 7, 8

[29] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, 2019. 1, 2, 5, 6, 7, 8

[30] 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. In *ECCV*, pages 388–404. Springer, 2022. 3, 5, 7, 8

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 2

[32] Zhaoyang Liu, Donghao Luo, Yabiao Wang, Limin Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Tong Lu. Teinet: Towards an efficient architecture for video recognition. In *AAAI*, pages 11669–11676, 2020. 1, 2, 5, 7, 8

[33] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *CVPR*, pages 3202–3211, 2022. 1, 3

[34] Zhaoyang Liu, Limin Wang, Wayne Wu, Chen Qian, and Tong Lu. Tam: Temporal adaptive module for video recognition. In *ICCV*, pages 13708–13718, 2021. 1, 2, 8

[35] Chenxu Luo and Alan L Yuille. Grouped spatial-temporal aggregation for efficient action recognition. In *ICCV*, pages 5512–5521, 2019. 2

[36] 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 and captioning. *Neurocomputing*, 508:293–304, 2022. 3

[37] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. In *ICCV*, pages 3163–3172, 2021. 3

[38] 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. In *ECCV*, pages 1–18. Springer, 2022. 3, 8

[39] Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. St-adapter: Parameter-efficient image-to-video transfer learning. In *NeurIPS*, 2022. 3, 5, 7, 8

[40] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatiotemporal representation with pseudo-3d residual networks. In *ICCV*, 2017. 1, 2

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 3, 5, 6, 8

[42] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021. 3, 6

[43] Michael S Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: What can 8 learned tokens do for images and videos? *arXiv preprint arXiv:2106.11297*, 2021. 3

[44] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022. 7

[45] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, pages 510–526. Springer, 2016. 5

[46] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014. 1, 2

[47] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Gate-shift networks for video action recognition. In *CVPR*, pages 1102–1111, 2020. 2

[48] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, pages 843–852, 2017. 3, 6, 8

[49] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023. 3, 6, 7

[50] Shuyang Sun, Zhanghui Kuang, Lu Sheng, Wanli Ouyang, and Wei Zhang. Optical flow guided feature: A fast and robust motion representation for video action recognition. In *CVPR*, 2018. 1, 2

[51] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 1, 2

[52] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *ICCV*, pages 5552–5561, 2019. 2

[53] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018. 1, 2

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. 2

[55] Heydi Méndez Vázquez, Edel García Reyes, and Yadira Condes Molleda. A new image division for lbp method to improve face recognition under varying lighting conditions. In *ICPR*, pages 1–4. IEEE, 2008. 2

[56] Heng Wang, Du Tran, Lorenzo Torresani, and Matt Feiszli. Video modeling with correlation networks. In *CVPR*, pages 352–361, 2020. 1, 2, 7

[57] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. Tdn: Temporal difference networks for efficient action recognition. In *CVPR*, pages 1895–1904, 2021. 1, 2, 5, 7, 8

[58] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016. 1, 2

[59] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *arXiv preprint arXiv:2109.08472*, 2021. 3

[60] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 1, 8

[61] Wenhao Wu, Dongliang He, Tianwei Lin, Fu Li, Chuang Gan, and Errui Ding. Mvfnet: Multi-view fusion network for efficient video recognition. In *AAAI*, 2021. 1, 2, 5, 7, 8

[62] Wenhao Wu, Dongliang He, Xiao Tan, Shifeng Chen, and Shilei Wen. Multi-agent reinforcement learning based frame sampling for effective untrimmed video recognition. In *ICCV*, 2019. 2

[63] Wenhao Wu, Dongliang He, Xiao Tan, Shifeng Chen, Yi Yang, and Shilei Wen. Dynamic inference: A new approach toward efficient video action recognition. In *Proceedings of CVPR Workshops*, pages 676–677, 2020. 2

[64] Wenhao Wu, Haipeng Luo, Bo Fang, Jingdong Wang, and Wanli Ouyang. Cap4video: What can auxiliary captions do for text-video retrieval? In *CVPR*, pages 10704–10713, 2023. 3

[65] Wenhao Wu, Zhun Sun, and Wanli Ouyang. Revisiting classifier: Transferring vision-language models for video recognition. In *AAAI*, pages 2847–2855, 2023. 3, 8

[66] Wenhao Wu, Zhun Sun, Yuxin Song, Jingdong Wang, and Wanli Ouyang. Transferring vision-language models for visual recognition: A classifier perspective. *International Journal of Computer Vision*, 2023. 3, 8

[67] Wenhao Wu, Xiaohan Wang, Haipeng Luo, Jingdong Wang, Yi Yang, and Wanli Ouyang. Bidirectional cross-modal knowledge exploration for video recognition with pretrained vision-language models. In *CVPR*, pages 6620–6630, 2023. 3, 8

[68] Wenhao Wu, Yuxiang Zhao, Yanwu Xu, Xiao Tan, Dongliang He, Zhikang Zou, Jin Ye, Yingying Li, Mingde Yao, Zichao Dong, et al. Dsanet: Dynamic segment aggregation network for video-level representation learning. *In Proc. ACMMM*, 2021. 1, 2

[69] Zuxuan Wu, Caiming Xiong, Chih-Yao Ma, Richard Socher, and Larry S Davis. Adaframe: Adaptive frame selection for fast video recognition. In *CVPR*, pages 1278–1287, 2019. 2

[70] Boyang Xia, Zhihao Wang, Wenhao Wu, Haoran Wang, and Jungong Han. Temporal saliency query network for efficient video recognition. In *ECCV*, pages 741–759, 2022. 2

[71] Boyang Xia, Wenhao Wu, Haoran Wang, Rui Su, Dongliang He, Haosen Yang, Xiaoran Fan, and Wanli Ouyang. Nsnet: Non-saliency suppression sampler for efficient video recognition. In *ECCV*, pages 705–723, 2022. 2

[72] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018. 1, 2

[73] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. In *CVPR*, pages 3333–3343, 2022. 3, 7, 8

[74] Taojiannan Yang, Yi Zhu, Yusheng Xie, Aston Zhang, Chen Chen, and Mu Li. Aim: Adapting image models for efficient video understanding. In *ICLR*, 2023. 3, 5, 7, 8

[75] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021. 3, 8

[76] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, pages 4694–4702, 2015. 1, 2

[77] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *CVPR*, pages 12104–12113, 2022. 3, 6

[78] Bowen Zhang, Limin Wang, Zhe Wang, Yu Qiao, and Hanli Wang. Real-time action recognition with enhanced motion vector cnns. In *CVPR*, 2016. 1, 2

[79] Bowen Zhang, Jiahui Yu, Christopher Fifty, Wei Han, Andrew M Dai, Ruoming Pang, and Fei Sha. Co-training transformer with videos and images improves action recognition. *arXiv preprint arXiv:2112.07175*, 2021. 3, 7

[80] Shiwen Zhang, Sheng Guo, Weilin Huang, Matthew R Scott, and Limin Wang. V4d: 4d convolutional neural networks for video-level representation learning. In *ICLR*, 2020. 2

[81] Xiao-Yu Zhang, Hai-Chao Shi, Chang-Sheng Li, and Li-Xin Duan. Twinnet: twin structured knowledge transfer network for weakly supervised action localization. *Machine Intelligence Research*, 19(3):227–246, 2022. 2

[82] Shuai Zhao, Linchao Zhu, Xiaohan Wang, and Yi Yang. Centerclip: Token clustering for efficient text-video retrieval. *The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022. 3

[83] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. Eco: Efficient convolutional network for online video understanding. In *ECCV*, 2018. 1