What Can Simple Arithmetic Operations Do for Temporal Modeling?
Supplementary Material

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

In this appendix, we provide additional details as follows: §A contains more implementation details, §B contains more ablation studies, §C contains additional results.

A. Implementation Details

A.1. Integration of ATM into Existing Backbones

Integration of ATM into ResNet [6]. As shown in Table A.1, we conduct experiments on two types of ResNet architectures: ResNet with basic blocks, and ResNet with bottleneck blocks. The ResNet basic block, which comprises two 3×3 convolutional layers, is utilized in ResNet-18/34. In contrast, ResNet-50/101/152 includes the ResNet bottleneck block, consisting of two 1×1 and one 3×3 convolutional layers. The ATM is integrated once after the last residual block of res_3, where the spatial resolution is 28×28, in both two ResNet architectures. To optimize computational efficiency, we temporarily reduce the resolution to 14×14 during the Context Spanning operation, followed by upsampling to 28×28 during Domain Transformation.

Integration of ATM into ViT [2]. We seamlessly integrate the ATM into two representative ViT architectures: ViT-Base and ViT-Large, as provided by CLIP [8]. These architectures are outlined in Table A.2. In order to capture temporal cues for each anchor frame, we incorporate the ATM after the multi-head attention module of the 7-th layer in ViT-B (which consists of 12 layers) and after the 14-th layer in ViT-L (which consists of 24 layers). To prepare the input for the ATM, we reshape the visual tokens within the ViT blocks, transforming them from one-dimensional spatial dimensions ($N = H \times W$) to two-dimensional spatial dimensions. As a result, we obtain $X^{\text{IN}} \in \mathbb{R}^{T\times C\times H\times W}$. We then apply the ATM directly to X^{IN}, similar to the way we apply it in ResNet.

A.2. Training Hyperparameters

All experiments are implemented in PyTorch. We use the configuration listed in Table A.3 unless otherwise specified.

stage	ResNet18	ResNet50	output sizes
raw clip	-	-	64×224^2
data layer	stride 8, 1^2	stride 8, 1^2	8×224^2
conv_1	1×7^2, 64 stride 1, 2^2	1×7^2, 64 stride 1, 2^2	8×112^2
pool_1	1×3^2 max stride 1, 2^2	1×3^2 max stride 1, 2^2	8×56^2
res_2	$\begin{bmatrix}1\times3^2, 64\\1\times3^2, 64\end{bmatrix}\times2$	$\begin{bmatrix}1\times1^2, 64\\1\times3^2, 64\\1\times1^2, 256\end{bmatrix}\times3$	8×56^2
res_3	$\begin{bmatrix}1\times3^2, 128\\1\times3^2, 128\end{bmatrix}\times2$	$\begin{bmatrix}1\times1^2, 128\\1\times3^2, 128\\1\times1^2, 512\end{bmatrix}\times4$	8×28^2
res_4	$\begin{bmatrix}1\times3^2, 256\\1\times3^2, 256\end{bmatrix}\times2$	$\begin{bmatrix}1\times1^2, 256\\1\times3^2, 256\\1\times1^2, 1024\end{bmatrix}\times6$	8×14^2
res_5	$\begin{bmatrix}1\times3^2, 512\\1\times3^2, 512\end{bmatrix}\times2$	$\begin{bmatrix}1\times1^2, 512\\1\times3^2, 512\\1\times1^2, 2048\end{bmatrix}\times3$	8×7^2
global average pool, fc			# classes

Table A.1. **Two backbones of the ResNet**. The dimensions of kernels are denoted by $\{T\times S^2, C\}$ for temporal, spatial, and channel sizes. Strides are denoted as $\{\text{temporal stride, spatial stride}^2\}$.

Model	Embedding dimension	Vision Transformer layers	width	heads
ViT-B/16	512	12	768	12
ViT-L/14	768	24	1024	16

Table A.2. Two backbones of the ViT.

B. Ablation Studies

Here, we present additional ablation studies conducted on the Something-Something V1 dataset, specifically using ResNet-18 as the backbone and utilizing 8 frames.

The Effect of Arithmetic Operations. In our ATM, we employ context spanning to generate L context frames for each frame, and then perform arithmetic operations between the features of the anchor frame and the L context frames to

Setting	ResNet		ViT			
	SSV1/V2	K400	SSV1/V2	K400	ANet	Charades
Optimization						
Batch size	64		256	256	256	256
Epochs	60	150	20 (B), 15 (L)	20 (B), 15 (L)	15	15
Optimizer	SGD		AdamW ($\beta_1 = 0.9, \beta_2 = 0.999$)			
Initial LR	0.01		7e-4	3e-4 (B), 2e-4 (L)	2e-4	2e-4
Layer Decay	-		0.7 (B), 0.75 (L)	0.65 (B), 0.7 (L)	0.7	0.7
LR Schedule	Multi-step, $\gamma = 0.1$		Cosine			
LR Steps	[30,45,55]	[80,120,140]	-			
Weight Decay	5e-4		0.05			
Linear Warm-Up	-		5 Epochs			
Pre-training	ImageNet-1K		WIT-400M			
Augmentation						
Training Resize	MultiScaleCrop		RandomSizedCrop			
Rand Augment	-		rand-m7-n4-mstd0.5-inc1	-	-	-
Random Flip	0.5		0.5			
Repeated Sampling	1		2			
Label Smoothing	-		0.1			
GrayScale	-		-	0.2	0.2	0.2
Mixup	-		0.8			
Cutmix	-		1.0			

Table A.3. Default training recipes. LR denotes the learning rate. B represents ViT-Base and L represents ViT-Large.

extract temporal clues. To evaluate the impact of arithmetic operations, we remove parameter-free arithmetic operations and directly use the features of context frames as temporal clues for the anchor frame. We find that this approach provides a basic level of temporal modeling. Next, we incorporate the outputs of arithmetic operations between the anchor frame and context frames as our temporal clues. We observe that different arithmetic operations yield varying degrees of improvement in temporal modeling, as shown in Table A.4.

Method	Top-1
w/o ATM	16.5%
ATM (Context features)	40.6%
ATM (✚)	41.1%
ATM (━)	43.6%
ATM (✖)	**46.5%**
ATM (➗)	43.1%

Table A.4. The effect of arithmetic operations. Backbone: R18.

Different Combinations of ATMs. In this part, we present several approaches for amalgamating ATMs (*e.g.*, ATM (✖) and ATM (━)). These approaches include (a) the cascade connection, (b) the parallel connection, and (c) the proposed ATM-style connection, as depicted in Figure A.1. We present the experimental results in Table A.5. The findings indicate that combining ATMs with either the cascade or parallel style leads to only marginal improvements over a single ATM (✖). This result emphasizes the importance of domain transformation operations that transform signals to a temporal-irrespective stem.

Combinations	Top-1
Single ATM (✖)	46.5%
Single ATM (━)	43.6%
(a) Cascade	46.6%
(b) Parallel	46.9%
(c) ATM-style	**48.2%**

Table A.5. Several combinations of ATMs. Backbone: R18.

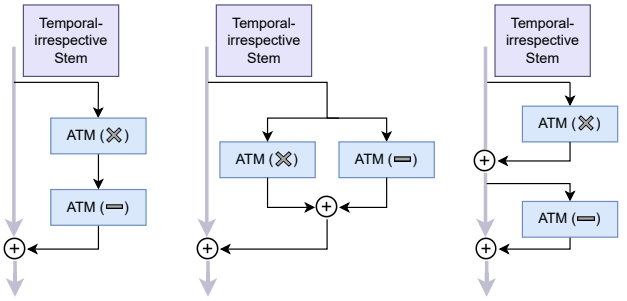
C. Additional Results

More Results on Kinetics-400, Something-Something V1 & V2. For readers' reference, we present our results with various views in Table A.7 and Table A.6.

Results on ActivityNet. To demonstrate the generalization ability of our method, we evaluate its performance on the widely-used untrimmed video benchmark, ActivityNet-v1.3 [1]. This dataset consists of 19,994 videos ranging from 5 to 10 minutes in length, covering 200 activity categories. We fine-tune the CLIP pre-trained ViT-L backbone with 16 frames on this dataset and report the top-1 accuracy

Method	Pretrain	Frame×Crops×Clip	GFLOPs	SSV1		SSV2	
				Top-1	Top-5	Top-1	Top-5
ATM ResNet50	ImageNet-1K	8×1×1	37×1	53.9%	81.8%	65.5%	89.9%
ATM ResNet50	ImageNet-1K	8×1×2	37×2	54.7%	82.6%	66.1%	90.2%
ATM ResNet50	ImageNet-1K	16×1×1	74×1	56.3%	83.4%	67.4%	91.1%
ATM ResNet50	ImageNet-1K	16×1×2	74×2	56.7%	83.6%	67.6%	91.2%
ATM ResNet50	ImageNet-1K	32×1×1	148×1	57.1%	84.0%	68.4%	91.5%
ATM ResNet50	ImageNet-1K	32×1×2	148×2	57.2%	84.3%	68.5%	91.6%
ATM ResNet50	ImageNet-1K	(8+16)×1×1	111×1	57.6%	84.4%	68.3%	91.6%
ATM ResNet50	ImageNet-1K	(8+16)×1×2	111×2	57.8%	84.8%	68.7%	91.7%
ATM ResNet50	ImageNet-1K	(8+16+32)×1×1	259×1	59.1%	85.7%	69.7%	92.4%
ATM ResNet101	ImageNet-1K	8×1×1	67×1	54.9%	82.3%	66.4%	90.3%
ATM ResNet101	ImageNet-1K	8×1×2	67×2	55.8%	82.9%	66.9%	90.7%
ATM ResNet101	ImageNet-1K	16×1×1	134×1	57.2%	84.1%	68.2%	91.5%
ATM ResNet101	ImageNet-1K	16×1×2	134×2	57.4%	84.4%	68.6%	91.6%
ATM ResNet101	ImageNet-1K	32×1×1	268×1	57.9%	84.3%	69.3%	92.0%
ATM ResNet101	ImageNet-1K	(8+16)×1×1	201×1	58.6%	84.9%	69.4%	92.1%
ATM ResNet101	ImageNet-1K	(8+16)×1×2	201×2	58.9%	85.1%	69.6%	92.3%
ATM ResNet101	ImageNet-1K	(8+16+32)×1×1	469×1	60.0%	86.1%	70.8%	92.9%
ATM ViT-B/16	WIT-400M	8×1×1	99×1	58.1%	84.7%	69.4%	92.2%
ATM ViT-B/16	WIT-400M	8×3×2	99×6	58.8%	85.4%	70.5%	92.7%
ATM ViT-B/16	WIT-400M	16×1×1	198×1	59.5%	86.1%	70.9%	92.8%
ATM ViT-B/16	WIT-400M	16×3×2	198×6	60.6%	86.5%	71.5%	93.0%
ATM ViT-B/16	WIT-400M	32×1×1	378×1	60.9%	85.9%	71.6%	93.2%
ATM ViT-B/16	WIT-400M	32×3×2	378×6	61.5%	86.2%	71.9%	93.3%
ATM ViT-L/14	WIT-400M	8×1×1	421×1	61.9%	87.0%	71.3%	93.1%
ATM ViT-L/14	WIT-400M	8×3×2	421×6	62.8%	87.6%	72.1%	93.6%
ATM ViT-L/14	WIT-400M	16×1×1	842×1	63.3%	87.5%	73.1%	93.5%
ATM ViT-L/14	WIT-400M	16×3×1	842×3	63.7%	88.0%	73.2%	93.7%
ATM ViT-L/14	WIT-400M	16×3×2	842×6	64.0%	88.0%	73.5%	93.7%
ATM ViT-L/14	Merged-2B	8×3×2	421×6	64.9%	88.9%	73.7%	94.1%
ATM ViT-L/14	Merged-2B	16×3×2	842×6	65.6%	88.6%	74.6%	94.4%

Table A.6. More results on Something-Something V1 & V2.



(a) Cascade (b) Parallel (c) ATM-style

Figure A.1. The different combinations of ATM(✕) and ATM(═).

and mean average precision (mAP) using official evaluation metrics. As shown in Table A.8, our method outperforms recent works, achieving an mAP accuracy of 94.7%.

Results on Charades. We also conduct experiments on the multi-label video recognition task using the Charades dataset [9]. This dataset consists of over 10,000 short video clips covering 157 action categories. We trained the CLIP pre-trained ViT-L backbone for this task and evaluated the results using the Mean Average Precision (mAP) metric. Table A.9 illustrates the effectiveness of our method in multi-label video classification.

References

[1] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 2

Method	Pretrain	Frame×Crops×Clip	GFLOPs	Top-1	Top-5
ATM ResNet50	ImageNet-1K	8×3×10	37×30	77.0%	92.9%
ATM ResNet50	ImageNet-1K	16×3×10	74×30	77.6%	93.2%
ATM ResNet101	ImageNet-1K	8×3×10	67×30	78.0%	93.5%
ATM ResNet101	ImageNet-1K	16×3×10	134×30	78.8%	93.7%
ATM ResNet152	ImageNet-1K	16×3×10	191×30	79.4%	93.7%
ATM R101+R152	ImageNet-1K	(16+16)×3×10	326×30	80.5%	94.4%
ATM ViT-B/16	WIT-400M	8×3×4	99×12	84.1%	96.3%
ATM ViT-L/14	WIT-400M	8×3×4	421×12	87.3%	97.4%
ATM ViT-L/14	WIT-400M	32×3×4	1684×12	88.0%	97.6%
ATM ViT-L/14 (336↑)	WIT-400M	32×3×4	3784×12	88.2%	97.9%
ATM ViT-L/14	Merged-2B	8×3×4	421×12	88.0%	97.6%
ATM ViT-L/14 (336↑)	Merged-2B	8×3×4	946×12	88.9%	97.8%
ATM ViT-L/14 (336↑)	Merged-2B	32×3×4	3784×12	89.4%	98.3%

Table A.7. More results on Kinetics-400.

Method	Top-1	mAP
ListenToLook [5]	-	89.9
MARL [13]	85.7	90.1
DSANet [14]	-	90.5
TSQNet [15]	88.7	93.7
Ours ViT-L	**90.2**	**94.7**

Table A.8. Comparisons with previous works on ActivityNet.

Method	**Frames**	**mAP**
MultiScale TRN [16]	-	25.2%
STM [7]	16	35.3%
Nonlocal [11]	-	37.5%
SlowFast R50 [4]	8+32	38.0%
SlowFast R101 [4]	16+64	42.5%
LFB+NL [12]	32	42.5%
X3D-XL (312↑) [3]	16	43.4%
ActionCLIP [10]	32	44.3%
Ours ViT-L	16	**48.5%**

Table A.9. Comparison with previous works on **Multi-Label** video dataset Charades.

[2] 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. 1

[3] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *CVPR*, pages 203–213, 2020. 4

[4] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. *ICCV*, 2019. 4

[5] Ruohan Gao, Tae-Hyun Oh, Kristen Grauman, and Lorenzo Torresani. Listen to look: Action recognition by previewing audio. In *CVPR*, pages 10457–10467, 2020. 4

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[7] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *ICCV*, pages 2000–2009, 2019. 4

[8] 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. 1

[9] 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. 3

[10] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *arXiv preprint arXiv:2109.08472*, 2021. 4

[11] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 4

[12] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *CVPR*, pages 284–293, 2019. 4

[13] 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. 4

[14] 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 aggre-

gation network for video-level representation learning. *In Proc. ACMMM*, 2021. 4

[15] 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. 4

[16] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, 2018. 4