Paper Review

Relational Self-Attention

What's Missing in Attention for Video Understanding

NeurIPS 2021

R&D Center (Industrial AI Research), POSCO ICT

Susang Kim

Contents

Introduction

Introduction

Surveillance cameras are widely being used in POSCO.
Real-time video understanding is an important step towards.
(strong spatio-temporal modeling & capacity at low latency)

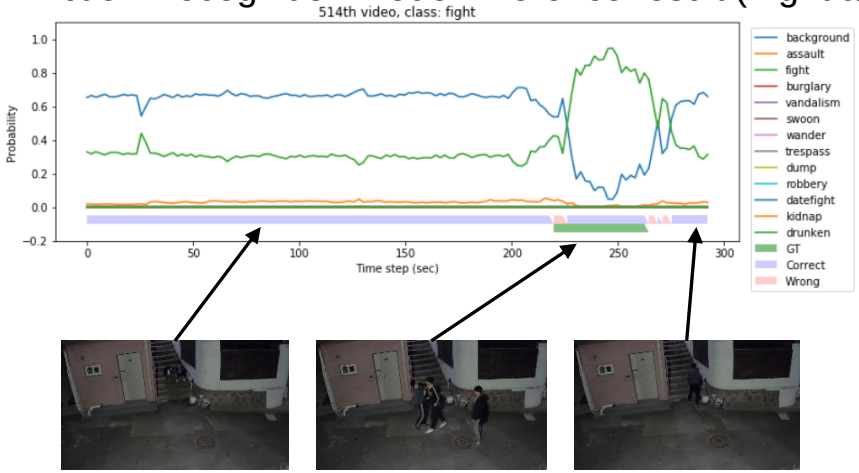
포스코ICT, `비전AI` 장착한 CCTV 내놓다

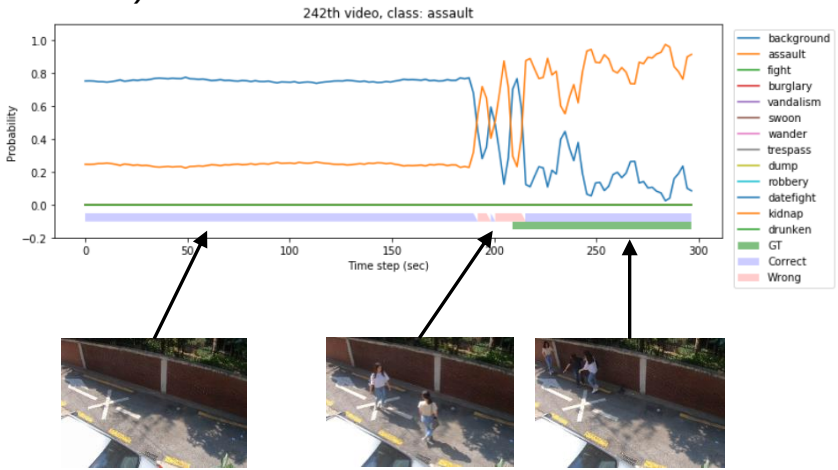
CCTV 영상에 포착된 행동 인식해 침입·방화시도, 작업자 이상행동 등 감지
AI 기반 영상분석 기술 KISA 인증 획득…안전·보안관리 등 산업현장 적용

안경애 기자 | 입력: 2021-10-27 10:52

http://www.dt.co.kr/contents.html?article_no=2021102702109931650006&ref=naver

Action Recognition Model Inference result (Fight and Assult)





Action Dataset : 12가지의 이상행동(폭행, 싸움, 절도, 기물파손, 실신, 배회, 침입, 투기, 강도, 데이트 폭력 및 추행, 납치, 주취행동), 총 700시간 (8400컷) 비디오 데이터셋 촬영 및 구축한 영상 데이터 제공) AI Hub : https://aihub.or.kr/aidata/139

Preliminaries - Representations for Video Classification

Hand-designed features : Wang et al., Action Recognition by Dense Trajectories, CVPR 2011.

Spatiotemporal ConvNets : Karpathy et al., Large-scale Video Classification with Convolutional Neural Networks, CVPR 2014

Two-stream ConvNets : K. Simonyan & A. Zisserman, Two-Stream Convolutional Networks for Action Recognition in Videos, NIPS 2014

3D ConvNets (C3D) : Du Tran et al., Learning spatiotemporal features with 3d convolutional networks. ICCV 2015

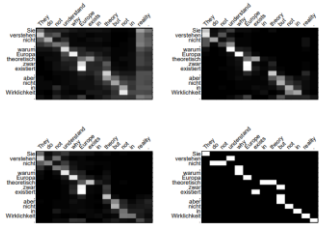
Temporal modeling without 3D ConvNets : X. Liu et al., TSM: Temporal Shift Module for Efficient Video Understanding, ICCV, 2019

Vision Transformer backbones : A. Arnab et al., ViViT: A Video Vision Transformer. ICCV 2021

Hybrid backbones(CNN+Self Attention UniFormer: Unified Transformer for Efficient Spatiotemporal Representation Learning. ICLR 2022**) :** K. Li et al.,

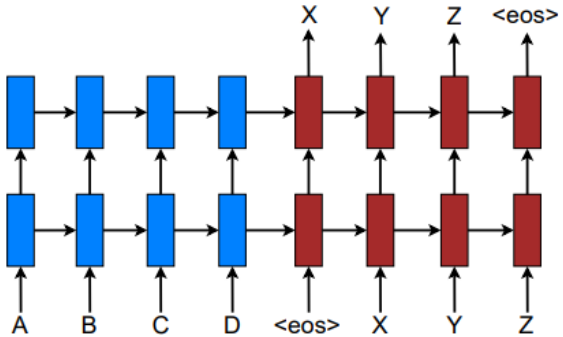
Attentional Mechanism(Neural machine translation)

Neural machine translation a stacking recurrent architecture for translating a source sequence.



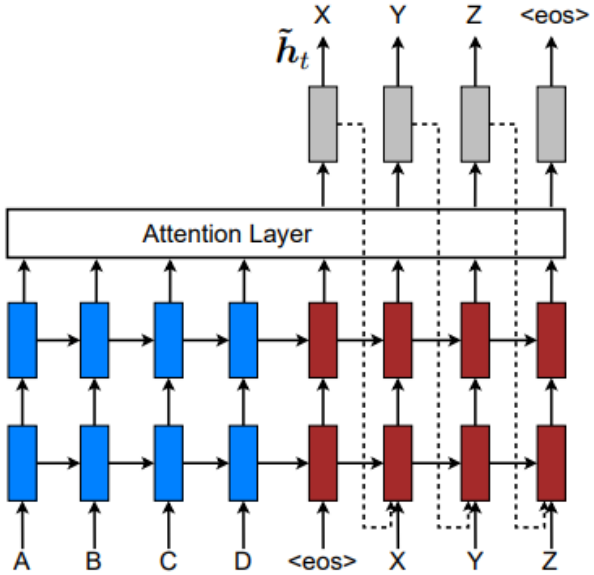
Attention Weights(Hard/Soft)

I like to order a pizza because I'm hungry



나는 지금 배가 고파 판교에 피자 주문하고 싶다

I like to order a pizza because I'm hungry



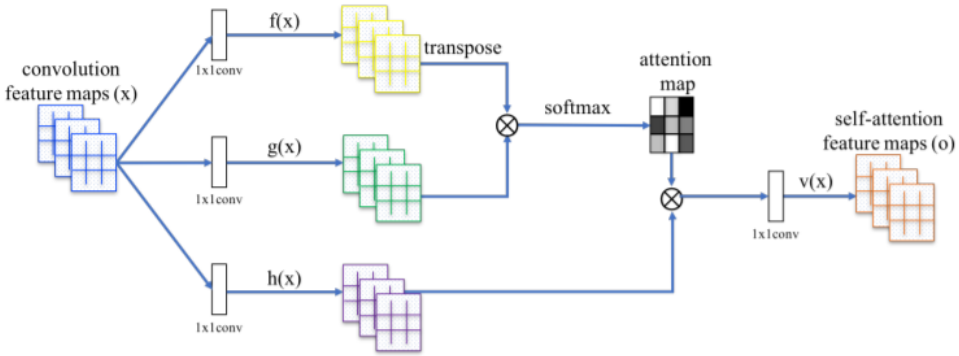
stacked multiple layers of an RNN(Attentional vectors)

나는 지금 배가 고파 판교에 피자 주문하고 싶다

Luong, Thang et al. "Effective Approaches to Attention-based Neural Machine Translation." EMNLP 2015.

Self-Attention

One is the total computational complexity per layer. Another is the amount of computation that can be parallelized, as measured by the minimum number of sequential operations required. The third is the path length between long-range dependencies in the network.
Learning long-range dependencies is a key challenge in many sequence transduction tasks. As side benefit, self-attention could yield more interpretable models. We inspect attention distributions from our models.



Attention(Q,K,V) : Self-Attention Generative Adversarial Networks

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

Attention Is All You Need [16]

Attention Is All You Need (NIPS 2017) [16]

Transformer, the first sequence transduction model based entirely on attention, replacing the recurrent layers.

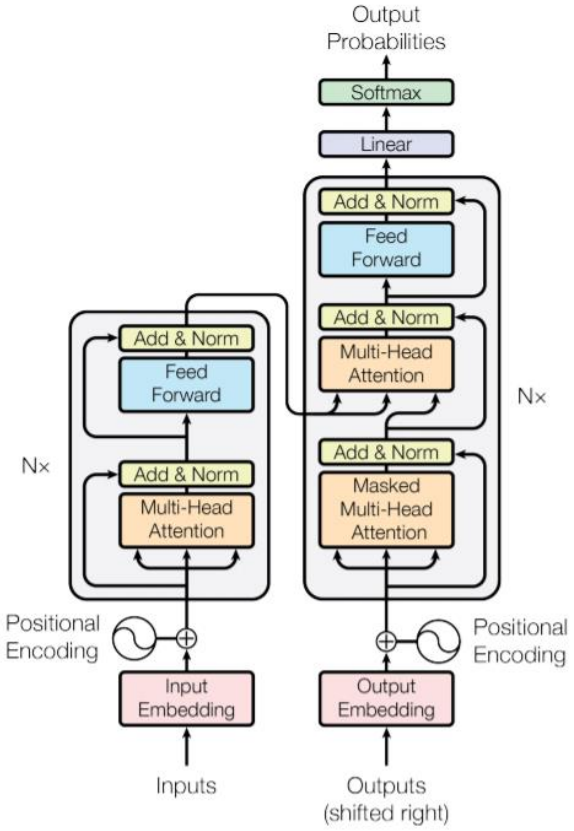
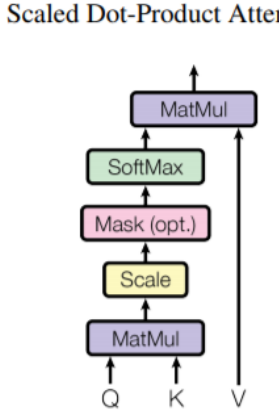
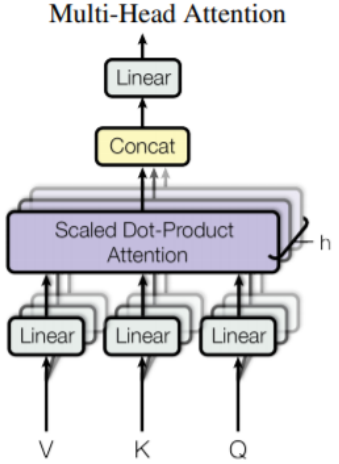


Figure 1: The Transformer - model architecture.





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

→ This lead to having more stable gradients

Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

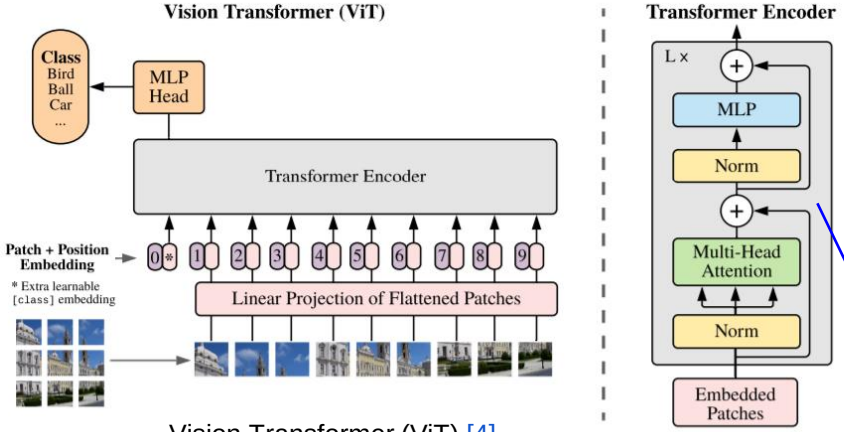
$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

Vision Transformer (ViT) (ICLR 2021)

"An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale"
CNNs is not necessary and a pure transformer applied directly to sequences of image patches
Vision Transformer has much less image-specific inductive bias than CNNs



Vision Transformer (ViT) [4]

Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification token" to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Table 1: Details of Vision Transformer model variants.

The sequence of linear embeddings of these patches as an input to a Transformer. The Transformer encoder consists of alternating layers of multi headed self attention (MSA) and MLP blocks. Layernorm (LN) is applied before every block, and residual connections after every block

$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}}; \mathbf{x}_p^1\mathbf{E}; \mathbf{x}_p^2\mathbf{E}; \cdots; \mathbf{x}_p^N\mathbf{E}] + \mathbf{E}_{pos}, \qquad \mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D},\ \mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D}$$
$$\mathbf{z}'_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \qquad \ell = 1 \ldots L$$
$$\mathbf{z}_\ell = \text{MLP}(\text{LN}(\mathbf{z}'_\ell)) + \mathbf{z}'_\ell, \qquad \ell = 1 \ldots L$$
$$\mathbf{y} = \text{LN}(\mathbf{z}_L^0)$$

$$[\mathbf{q}, \mathbf{k}, \mathbf{v}] = \mathbf{z}\mathbf{U}_{qkv} \qquad \mathbf{U}_{qkv} \in \mathbb{R}^{D \times 3D_h},$$
$$A = \text{softmax}\left(\mathbf{q}\mathbf{k}^\top / \sqrt{D_h}\right) \qquad A \in \mathbb{R}^{N \times N},$$
$$\text{SA}(\mathbf{z}) = A\mathbf{v}.$$

$$\text{MSA}(\mathbf{z}) = [\text{SA}_1(z); \text{SA}_2(z); \cdots; \text{SA}_k(z)]\,\mathbf{U}_{msa} \qquad \mathbf{U}_{msa} \in \mathbb{R}^{k \cdot D_h \times D}$$

The Vision Transformer performs well when pre-trained on a large JFT-300M dataset. With fewer inductive biases for vision than ResNets.
※ JFT-300M Dataset (300M web images / ~ 19K categories)

On Layer Normalization in the Transformer Architecture (ICML 2020)

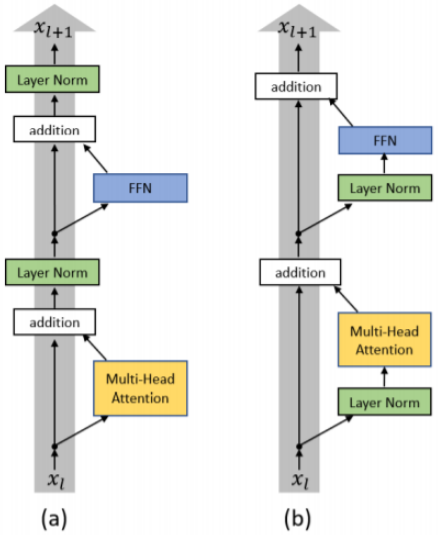
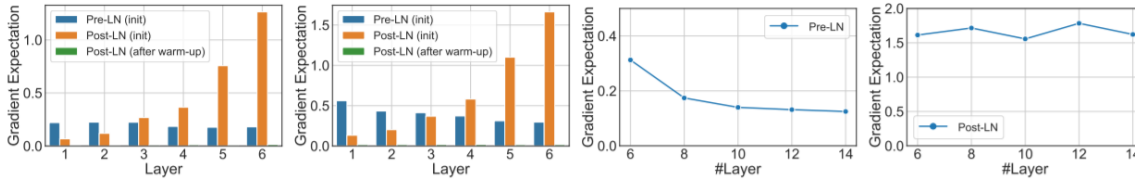


Figure 1. (a) Post-LN Transformer layer; (b) Pre-LN Transformer layer.

Table 1. Post-LN Transformer v.s. Pre-LN Transformer

Post-LN Transformer	Pre-LN Transformer
$x_{l,i}^{post,1} = \text{MultiHeadAtt}(x_{l,i}^{post}, [x_{l,1}^{post}, \cdots, x_{l,n}^{post}])$	$x_{l,i}^{pre,1} = \text{LayerNorm}(x_{l,i}^{pre})$
$x_{l,i}^{post,2} = x_{l,i}^{post} + x_{l,i}^{post,1}$	$x_{l,i}^{pre,2} = \text{MultiHeadAtt}(x_{l,i}^{pre,1}, [x_{l,1}^{pre,1}, \cdots, x_{l,n}^{pre,1}])$
$x_{l,i}^{post,3} = \text{LayerNorm}(x_{l,i}^{post,2})$	$x_{l,i}^{pre,3} = x_{l,i}^{pre} + x_{l,i}^{pre,2}$
$x_{l,i}^{post,4} = \text{ReLU}(x_{l,i}^{post,3} W^{1,l} + b^{1,l}) W^{2,l} + b^{2,l}$	$x_{l,i}^{pre,4} = \text{LayerNorm}(x_{l,i}^{pre,3})$
$x_{l,i}^{post,5} = x_{l,i}^{post,3} + x_{l,i}^{post,4}$	$x_{l,i}^{pre,5} = \text{ReLU}(x_{l,i}^{pre,4} W^{1,l} + b^{1,l}) W^{2,l} + b^{2,l}$
$x_{l+1,i}^{post} = \text{LayerNorm}(x_{l,i}^{post,5})$	$x_{l+1,i}^{pre} = x_{l,i}^{pre,5} + x_{l,i}^{pre,3}$
	Final LayerNorm: $x_{Final,i}^{pre} \leftarrow \text{LayerNorm}(x_{L+1,i}^{pre})$



(a) W^1 in the FFN sub-layers (b) W^2 in the FFN sub-layers (c) Pre-LN Transformer (d) Post-LN Transformer

Figure 3. The norm of gradients of 1. different layers in the 6-6 Transformer (a,b). 2. $W^{2,L}$ in different size of the Transformer (c,d).

Theorem 1 (Gradients of the last layer in the Transformer). *Assume that $\|x_{L,i}^{post,5}\|_2^2$ and $\|x_{L+1,i}^{pre}\|_2^2$ are (ϵ, δ)-bounded for all i, where ϵ and $\delta = \delta(\epsilon)$ are small numbers. Then with probability at least $0.99 - \delta - \frac{\epsilon}{0.9+\epsilon}$, for the Post-LN Transformer with L layers, the gradient of the parameters of the last layer satisfies*

$$\|\frac{\partial \tilde{\mathcal{L}}}{\partial W^{2,L}}\|_F \leq \mathcal{O}(d\sqrt{\ln d})$$

Pre-LN Transformer with L layers,

$$\|\frac{\partial \tilde{\mathcal{L}}}{\partial W^{2,L}}\|_F \leq \mathcal{O}\left(d\sqrt{\frac{\ln d}{L}}\right).$$

$$\text{lr}(t) = \frac{t}{T_{\text{warmup}}} \text{lr}_{max}, t \leq T_{\text{warmup}}.$$

In this paper, we study why the learning rate warm-up stage is important in training the Transformer and show that the location of layer normalization matters.

Learning Deep Transformer Models for Machine Translation https://arxiv.org/pdf/1906.01787.pdf

DeiT (Data-efficient image Transformers) (ICML 2021)

"Training data-efficient image transformers & distillation through attention"
DeiT are image transformers that do not require very large amount of data to be trained.
we train a vision transformer on a single 8-GPU node in two to three days that is competitive with convnets having a similar number of parameters and efficiency.

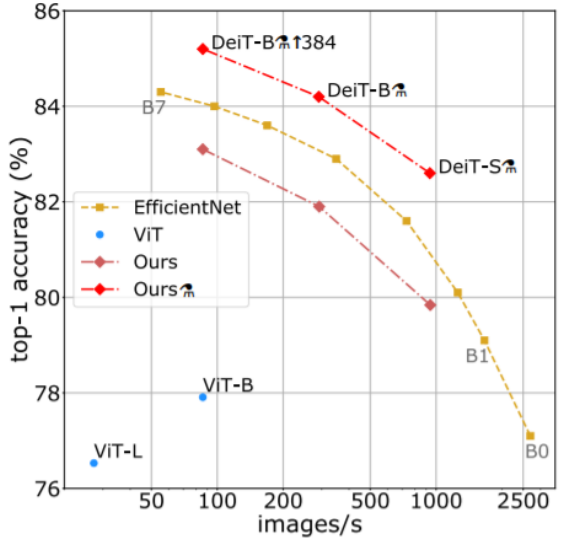


Figure 1: Throughput and accuracy on Imagenet of our methods compared to EfficientNets, trained on Imagenet1k only. The throughput is measured as the number of images processed per second on a V100 GPU. DeiT-B is identical to VIT-B, but the training is more adapted to a data-starving regime. It is learned in a few days on one machine. The symbol ⚗ refers to models trained with our transformer-specific distillation. See Table 5 for details and more models.

$$\mathcal{L}_{\text{global}} = (1-\lambda)\mathcal{L}_{\text{CE}}(\psi(Z_s), y) + \lambda\tau^2\text{KL}(\psi(Z_s/\tau), \psi(Z_t/\tau)).$$

$$\mathcal{L}^{\text{hardDistill}}_{\text{global}} = \frac{1}{2}\mathcal{L}_{\text{CE}}(\psi(Z_s), y) + \frac{1}{2}\mathcal{L}_{\text{CE}}(\psi(Z_s), y_t).$$

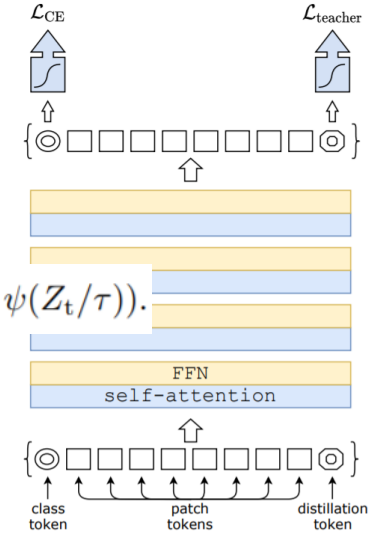
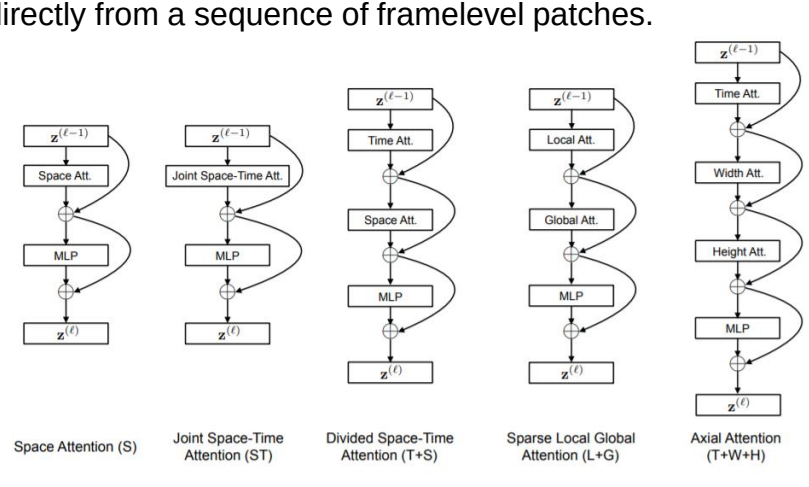


Table 7: We compare Transformers based models on different transfer learning task with ImageNet pre-training. We also report results with convolutional architectures for reference.

Model	ImageNet	CIFAR-10	CIFAR-100	Flowers	Cars	iNat-18	iNat-19	im/sec
Grafit ResNet-50 [49]	79.6	-	-	98.2	92.5	69.8	75.9	1226.1
Grafit RegNetY-8GF [49]	-	-	-	99.0	94.0	76.8	80.0	591.6
ResNet-152 [10]	-	-	-	-	-	69.1	-	526.3
EfficientNet-B7 [48]	84.3	98.9	91.7	98.8	94.7	-	-	55.1
ViT-B/32 [15]	73.4	97.8	86.3	85.4	-	-	-	394.5
ViT-B/16 [15]	77.9	98.1	87.1	89.5	-	-	-	85.9
ViT-L/32 [15]	71.2	97.9	87.1	86.4	-	-	-	124.1
ViT-L/16 [15]	76.5	97.9	86.4	89.7	-	-	-	27.3
DeiT-B	81.8	99.1	90.8	98.4	92.1	73.2	77.7	292.3
DeiT-B↑384	83.1	99.1	90.8	98.5	93.3	79.5	81.4	85.9
DeiT-B⚗	83.4	99.1	91.3	98.8	92.9	73.7	78.4	290.9
DeiT-B⚗↑384	84.4	99.2	91.4	98.9	93.9	80.1	83.0	85.9

Is Space-Time Attention All You Need for Video Understanding? (ICML 2021)

"TimeSformer" adapts the standard Transformer architecture to video by enabling spatiotemporal feature learning directly from a sequence of framelevel patches.



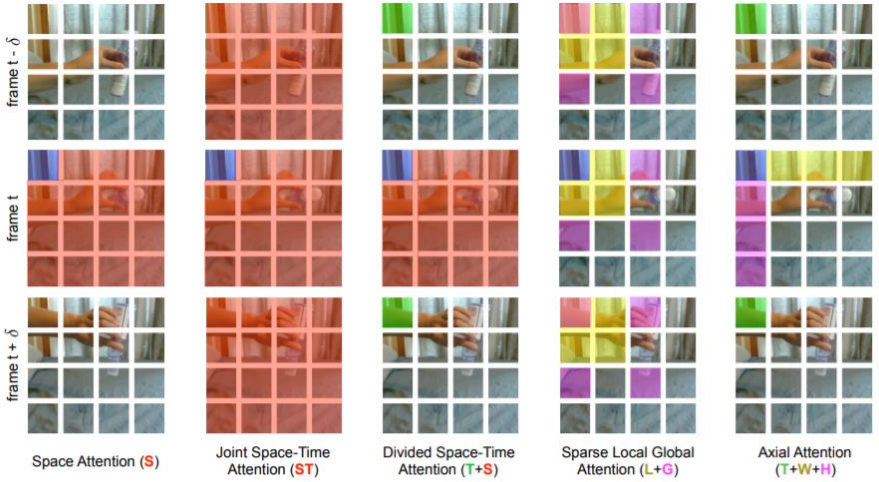
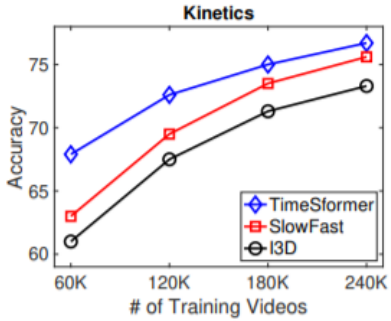


Figure 1. The video self-attention blocks that we investigate in this work. Each attention layer implements self-attention (Vaswani et al., 2017b) on a specified spatiotemporal neighborhood of frame-level patches (see Figure 2 for a visualization of the neighborhoods). We use residual connections to aggregate information from different attention layers within each block. A 1-hidden-layer MLP is applied at the end of each block. The final model is constructed by repeatedly stacking these blocks on top of each other.

Model	Pretrain	K400 Training Time (hours)	K400 Acc.	Inference TFLOPs	Params
I3D 8x8 R50	ImageNet-1K	444	71.0	1.11	28.0M
I3D 8x8 R50	ImageNet-1K	1440	73.4	1.11	28.0M
SlowFast R50	ImageNet-1K	448	70.0	1.97	34.6M
SlowFast R50	ImageNet-1K	3840	75.6	1.97	34.6M
SlowFast R50	N/A	6336	76.4	1.97	34.6M
TimeSformer	ImageNet-1K	**416**	75.8	**0.59**	121.4M
TimeSformer	ImageNet-21K	**416**	**78.0**	**0.59**	121.4M

Table 2. Comparing TimeSformer to SlowFast and I3D. We observe that TimeSformer has lower inference cost despite having a larger number of parameters. Furthermore, the cost of training TimeSformer on video data is much lower compared to SlowFast and I3D, even when all models are pretrained on ImageNet-1K.



Sequence of frame-level patches with a size of 16 × 16 pixels. query patch and show in non-blue colors its self-attention space-time neighborhood under each scheme. Patches without color are not used for the self-attention computation of the blue patch. Multiple colors within a scheme denote attentions separately applied along different dimensions (e.g., space and time for (T+S)) or over different neighborhoods (e.g., for (L+G)). Note that self-attention is computed for every single patch in the video clip, every patch serves as a query. it extends in the same fashion to all frames of the clip

ViViT: A Video Vision Transformer (ICCV 2021)

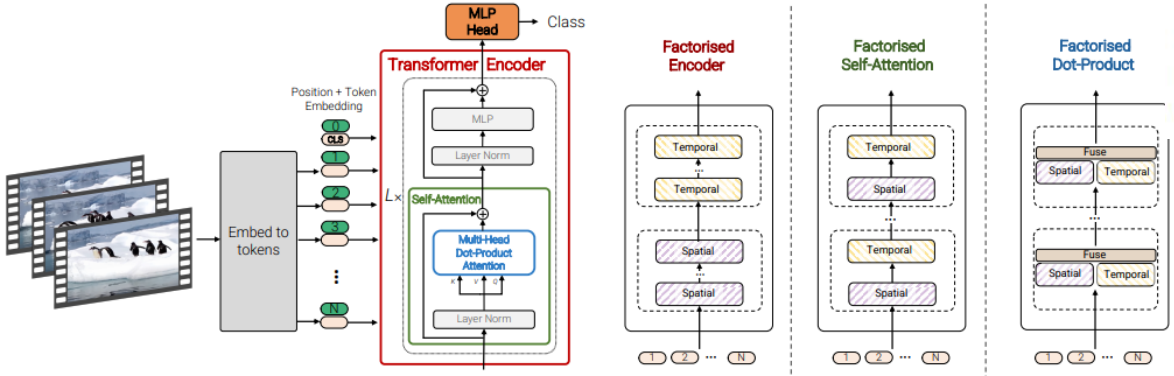


Figure 1: We propose a pure-transformer architecture for video classification, inspired by the recent success of such models for images [15]. To effectively process a large number of spatio-temporal tokens, we develop several model variants which factorise different components of the transformer encoder over the spatial- and temporal-dimensions. As shown on the right, these factorisations correspond to different attention patterns over space and time.

We consider two simple methods for mapping a video $\mathbf{V} \in \mathbb{R}^{T \times H \times W \times C}$ to a sequence of tokens $\tilde{\mathbf{z}} \in \mathbb{R}^{n_t \times n_h \times n_w \times d}$. We then add the positional embedding and reshape into $\mathbb{R}^{N \times d}$ to obtain $\mathbf{z}$, the input to the transformer.

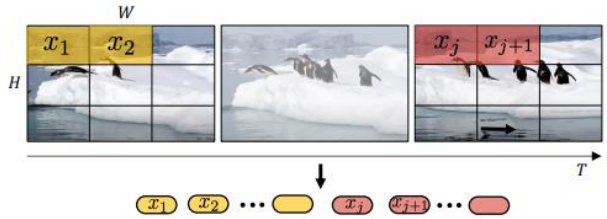


Figure 2: Uniform frame sampling: We simply sample n_t frames, and embed each 2D frame independently following ViT [15].

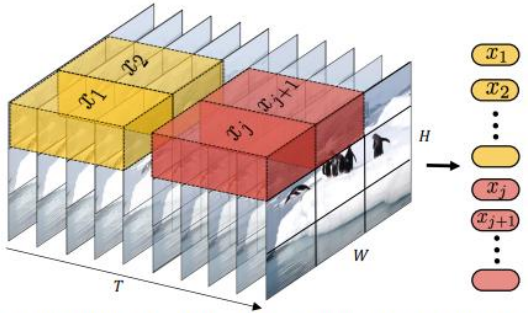


Figure 3: Tubelet embedding. We extract and linearly embed non-overlapping tubelets that span the spatio-temporal input volume.

Pure-transformer based models for video classification, we propose several, efficient variants of our model which factorise the spatial- and temporal-dimensions of the input. how we can effectively regularise the model during training and leverage pretrained image models to be able to train on comparatively small datasets.

Model 1: Spatio-temporal attention : all spatio-temporal tokens
Model 2: Factorised encoder : two separate transformer encoders
Model 3: Factorised self-attention : the order of spatial-then-temporal selfattention
Model 4: Factorised dot-product attention : $\mathbf{Y} = \mathrm{Concat}(\mathbf{Y}_s, \mathbf{Y}_t)\mathbf{W}_O$

$t \times h \times w, n_t = \lfloor \frac{T}{t} \rfloor, n_h = \lfloor \frac{H}{h} \rfloor$ and $n_w = \lfloor \frac{W}{w} \rfloor$

Multiscale Vision Transformers(MViT) (ICCV 2021)

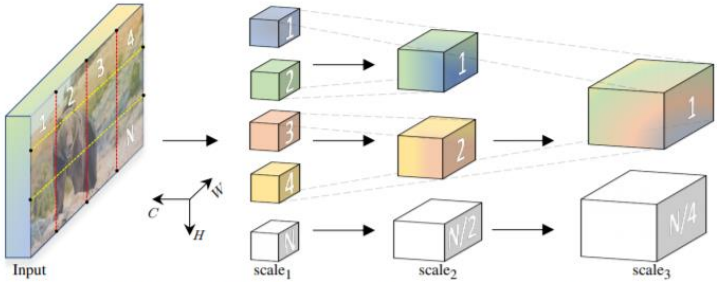
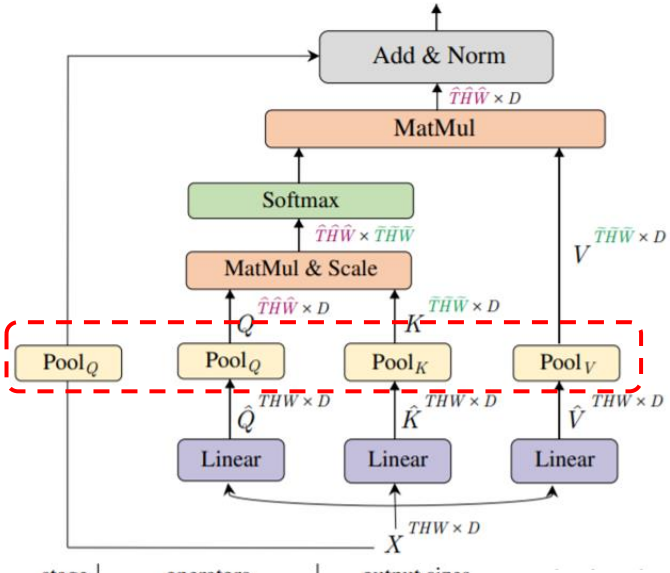


Figure 1. **Multiscale Vision Transformers** learn a hierarchy from *dense* (in space) and *simple* (in channels) to *coarse* and *complex* features. Several resolution-channel *scale* stages progressively *increase* the channel capacity of the intermediate latent sequence while *reducing* its length and thereby spatial resolution.



stage	operators	output sizes
data	stride 8×1×1	8×224×224
patch$_1$	1×16×16, 768 stride 1×16×16	768×8×14×14
scale$_2$	MHA(768) MLP(3072) ×12	768×8×14×14

(a) **ViT**-B with **179.6**G FLOPs, **87.2**M param, **16.8**G memory, and **68.5**% top-1 accuracy.

stage	operators	output sizes
data	stride 4×1×1	16×224×224
cube$_1$	3×7×7, 96 stride 2×4×4	96×8×56×56
scale$_2$	MHPA(96) MLP(384) ×1	96×8×56×56
scale$_3$	MHPA(192) MLP(768) ×2	192×8×28×28
scale$_4$	MHPA(384) MLP(1536) ×11	384×8×14×14
scale$_5$	MHPA(768) MLP(3072) ×2	768×8×7×7

(b) **MViT**-B with **70.5**G FLOPs, **36.5**M param, **6.8**G memory, and **77.2**% top-1 accuracy.

stage	operators	output sizes
data	stride 4×1×1	16×224×224
cube$_1$	3×8×8, 128 stride 2×8×8	128×8×28×28
scale$_2$	MHPA(128) MLP(512) ×3	128×8×28×28
scale$_3$	MHPA(256) MLP(1024) ×7	256×8×14×14
scale$_4$	MHPA(512) MLP(2048) ×6	512×8×7×7

(c) **MViT**-S with **32.9**G FLOPs, **26.1**M param, **4.5**G memory, and **74.3**% top-1 accuracy.

Table 3. Comparing **ViT**-B to two instantiations of **MViT** with varying complexity, **MViT**-S in (c) and **MViT**-B in (b). **MViT**-S operates at a lower spatial resolution and lacks a first high-resolution stage. The top-1 accuracy corresponds to *5-Center* view testing on K400. FLOPs correspond to a single inference clip, and memory is for a training batch of 4 clips. See Table 2 for the general **MViT**-B structure.

Paper Review

Relational Self-Attention (NeurIPS 2021)

Relational Self-Attention: What's Missing in Attention for Video Understanding

Manjin Kim[1]* Heeseung Kwon[1]* Chunyu Wang[2] Suha Kwak[1] Minsu Cho[1]
[1]POSTECH [2]Microsoft Research Asia
http://cvlab.postech.ac.kr/research/RSA/

Abstract

Convolution has been arguably the most important feature transform for modern neural networks, leading to the advance of deep learning. Recent emergence of Transformer networks, which replace convolution layers with self-attention blocks, has revealed the limitation of stationary convolution kernels and opened the door to the era of dynamic feature transforms. The existing dynamic transforms, including self-attention, however, are all limited for video understanding where correspondence relations in space and time, *i.e.*, motion information, are crucial for effective representation. In this work, we introduce a relational feature transform, dubbed the *relational self-attention (RSA)*, that leverages rich structures of spatio-temporal relations in videos by dynamically generating relational kernels and aggregating relational contexts. Our experiments and ablation studies show that the RSA network substantially outperforms convolution and self-attention counterparts, achieving the state of the art on the standard motion-centric benchmarks for video action recognition, such as Something-Something-V1&V2, Diving48, and FineGym.

A new dynamic feature transform, which effectively captures both visual appearance and spatio-temporal motion dynamics for video understanding.
[Submitted on 2 Nov 2021]

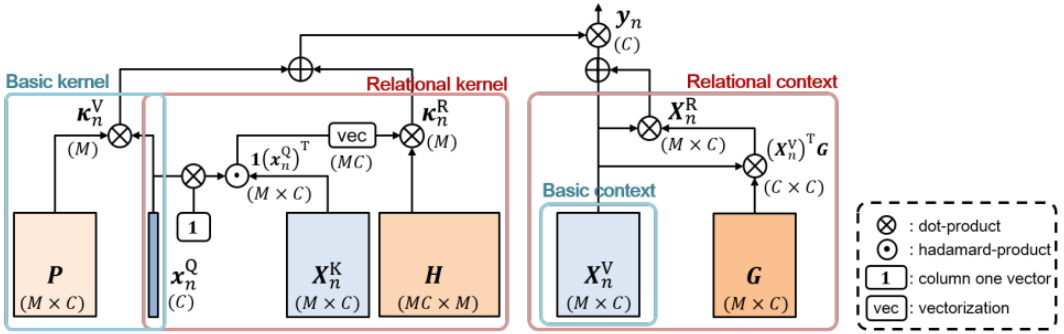
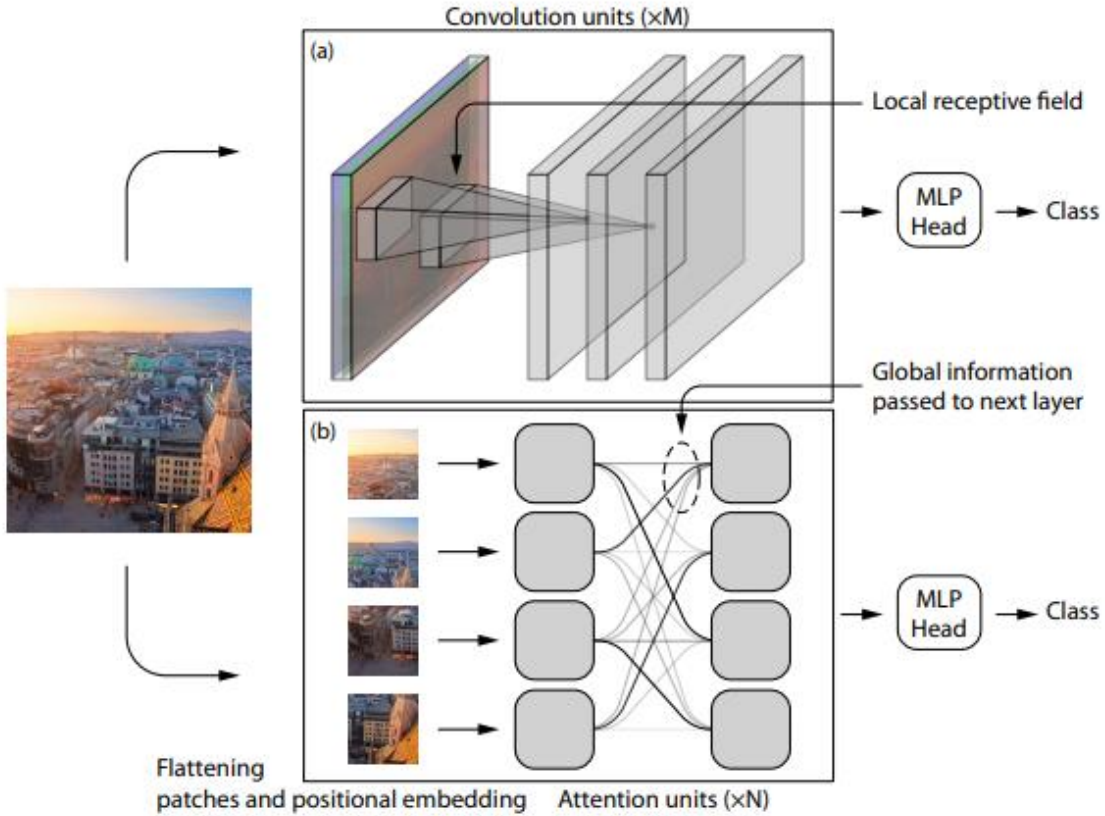


Figure 2: **Computational graph of RSA.** RSA consists of two types of kernels (basic and relational kernel) and two contexts (basic and relational context). See text for details.

Re-interpret dynamic feature transforms in a unified way, and provide in-depth analysis on their capability of learning video representation, designed to learn rich spatio-temporal interaction patterns across input contents.

Convolution vs. Transformer

It is difficult for ConvNets to capture long-term dependencies, while self-attention layers are global.



Convolution is efficient in memory and compute.
Local connectivity can lead to loss of global context.
Bad at long sequences(Need to stack many conv layers for outputs to “see” the whole sequence).

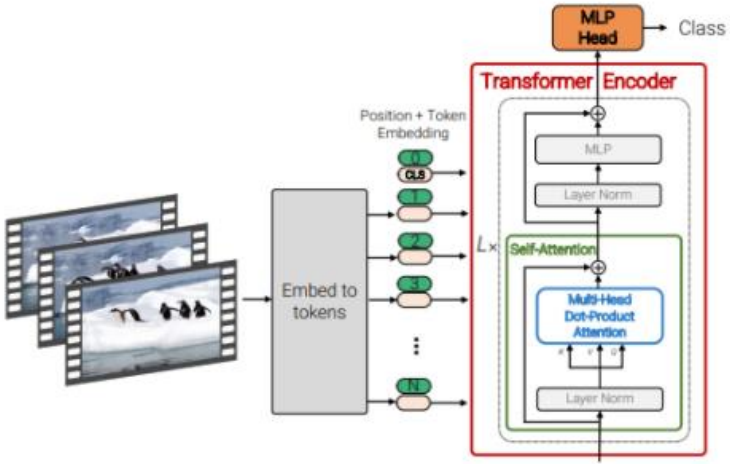
Transformers are flexible and attend to information at various distances away from Patch.
Good at long sequences
- output sees “all” inputs.
Dynamic w.r.t input
- output “sees” inputs adaptively.

Very memory-intensive

S Tuli et al., “Are Convolutional Neural Networks or Transformers more like human vision?,” CogSci, 2021.

Motivation - Problems in Self-Attention for learning motion

Vision Transformer, the vision transduction model based entirely on attention, replacing the CNN layers.



Video Vision Transformer (ViViT) [1]

$\boldsymbol{q}_n \in \mathbb{R}^C$
$\boldsymbol{K}_n, \boldsymbol{V}_n \in \mathbb{R}^{M \times C}$
$\boldsymbol{P} \in \mathbb{R}^{C \times M}$

$$\boldsymbol{y}_{\mathrm{n}} = softmax\left(\boldsymbol{q}_n \boldsymbol{K}_n^{\mathrm{T}} + \boldsymbol{q}_n \boldsymbol{P}^{\mathrm{T}}\right)\boldsymbol{V}_n$$

approach	model	K400 top-1(%)(a)	SSV2 top-1(%)(b)	Gap(%) (a)-(b)
ConvNet	TSM [12]	74.1	**63.4**	10.7
	MSNet [8]	76.4	**64.7**	11.7
	MoViNet-A3 [7]	78.2	**64.1**	14.1
	SELFYNet [9]	76.9	**65.7**	11.2
Transformer	TimeSformer-L [2]	**80.7**	62.3	18.4
	MViT-B 16x4 [5]	**78.4**	64.7	13.7
	ViViT-L [1]	**81.7**	65.9	15.8
	Swin-B [17]	**82.7**	69.6	13.1

Table 1. Performance comparison on Kinetics-400 (K400) and Something-V2 (SSV2)

Self-attention [16] represents the global information of a sequence.
(global view - appearance is better)

Convolution kernel is spatial-agnostic and channel-specific.
(local view - motion is better)

Attention is NOT All You Need, at least, for Videos.
(no improvement over generic motion information)

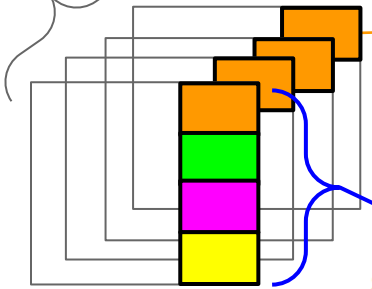
Feature transform

M : the size of neighborhood,
n : a specific position(spatio-temporal)
CxC : channel dimension (in x out)
E : learnable embedding matrices

Basic Context

$$\boldsymbol{E}^{\mathrm{Q}}, \boldsymbol{E}^{\mathrm{K}}, \boldsymbol{E}^{\mathrm{V}} \in \mathbb{R}^{C \times C} \longrightarrow \boldsymbol{X}_n^{\mathrm{V}} = \boldsymbol{X}_n \boldsymbol{E}^{\mathrm{V}}$$

$$\boldsymbol{X}_n \in \mathbb{R}^{M \times C}$$



$$\mathbf{1} \in \mathbb{R}^{M}$$

$$\mathbf{1}(\boldsymbol{x}_n^{\mathrm{Q}})^{\mathsf{T}} \odot \boldsymbol{X}_n^{\mathrm{K}} \in \mathbb{R}^{M \times C}$$

M(e.g., M=5×7×7)

$$\boldsymbol{X} \in \mathbb{R}^{T \times H \times W \times C}$$

$$\{\boldsymbol{x}_{t',h',w'}\}_{(t',h',w') \in \mathcal{N}_{(t,h,w)}}$$

$$\boldsymbol{x}_n \in \mathbb{R}^{C} \longrightarrow \boldsymbol{x}_n^{\mathrm{Q}} = \boldsymbol{x}_n \boldsymbol{E}^{\mathrm{Q}}$$

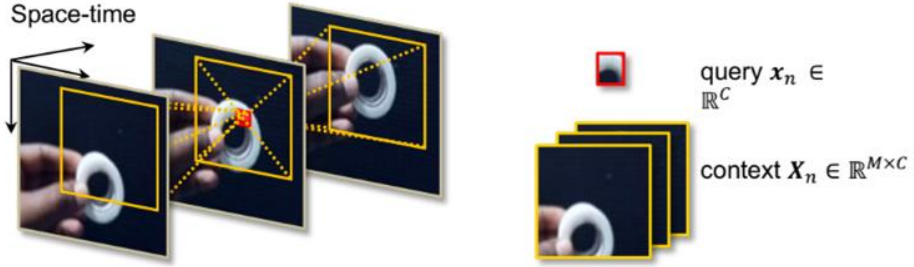
Target Vector (Channel dimension)

$$\boldsymbol{X}_n \colon \boldsymbol{y}_n = f(\boldsymbol{x}_n, \boldsymbol{X}_n; \boldsymbol{W})$$

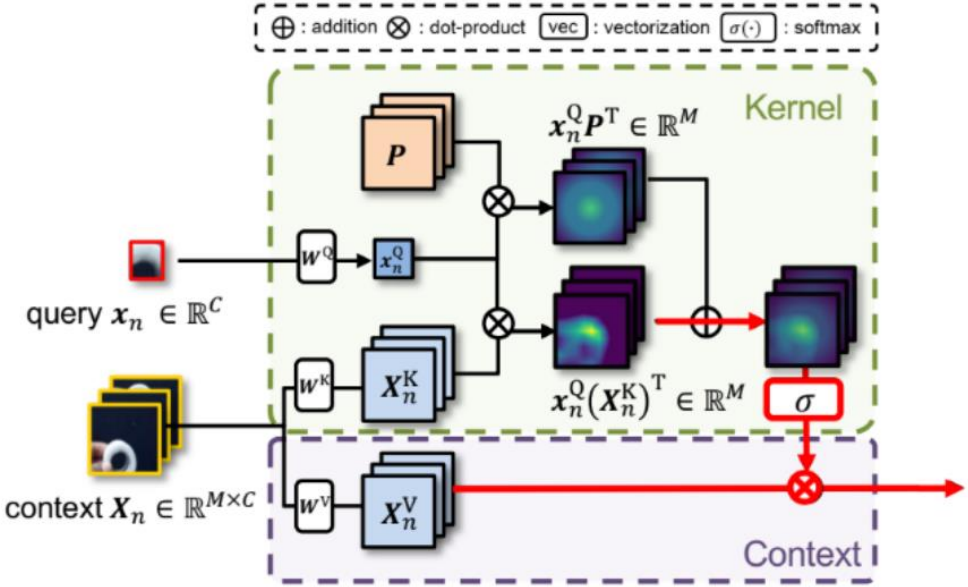
Output Vector

$$\boldsymbol{y}_n \in \mathbb{R}^{C}$$

Self-Attention in Space and Time



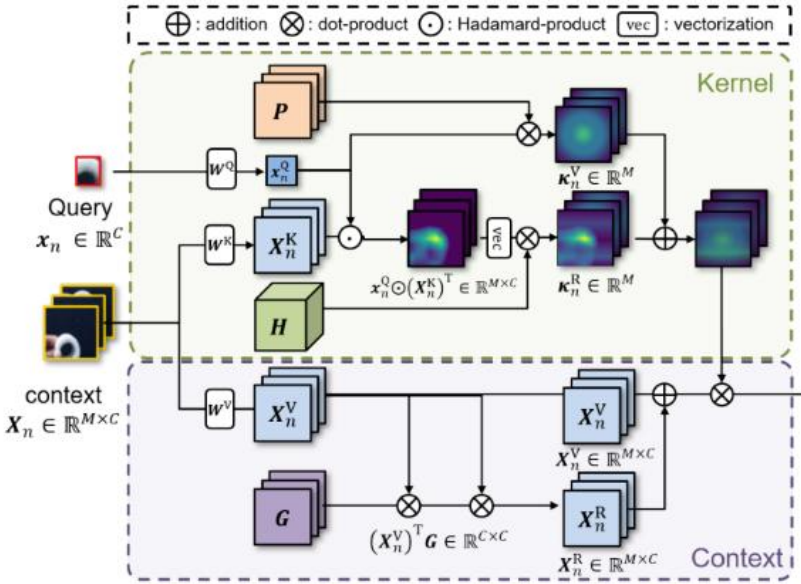
Individual query-key interaction for each kernel weight.
Permutation-invariant output (motion-agnostic)
Limited expressive ability due to the softmax



$y_n \in \mathbb{R}^C$
$x_n^{\mathrm{Q}} \in \mathbb{R}^C$
$X_n^{\mathrm{K}}, X_n^{\mathrm{V}}, P \in \mathbb{R}^{M \times C}$
$P \in \mathbb{R}^{M \times C}$
$\sigma(\cdot)$: softmax

$$y_n = \underbrace{\sigma(x_n^{\mathrm{Q}} (X_n^{\mathrm{K}})^{\mathrm{T}} + x_n^{\mathrm{Q}} P^{\mathrm{T}})}_{\text{Kernel}} \underbrace{X_n^{\mathrm{V}}}_{\text{Context}}$$

Relational Self-Attention (RSA)



$$\kappa_n^{\mathrm{V}} = x_n^{\mathrm{Q}} P^{\mathrm{T}}$$

Basic Kernel : Weights based on the content of the query itself

$$\kappa_n^{\mathrm{R}} = \mathrm{vec}(\mathbf{1}(x_n^{\mathrm{Q}})^{\mathrm{T}} \odot (X_n^{\mathrm{K}})^{\mathrm{T}}) H$$

Relational Kernel: Weights based on the query-key interactions

$$y_{\mathrm{n}} = \underbrace{(\kappa_n^{\mathrm{V}} + \kappa_n^{\mathrm{R}})}_{\text{Kernel}} \underbrace{(X_n^{\mathrm{V}} + X_n^{\mathrm{R}})}_{\text{Context}}$$

$$X_n^{\mathrm{V}} \in \mathbb{R}^{M\times C}$$

Basic Context : Appearance information of the context
(Equivalent to value embeddings)

$$x_n^{\mathrm{Q}} \in \mathbb{R}^C$$
$$X_n^{\mathrm{K}}, X_n^{\mathrm{V}} \in \mathbb{R}^{M\times C}$$
$$H \in \mathbb{R}^{MC\times M}$$
$$P \in \mathbb{R}^{M\times C}$$
$$G \in \mathbb{R}^{M\times C}$$

$$x_n^{\mathrm{R}} = x_n^{\mathrm{V}} (X_n^{\mathrm{V}})^{\mathrm{T}} G$$

Relational context : Self-correlation of the basic context
- motion information of the context

Datasets - Motion centric benchmark

Something-something v1 & v2 (SS-V1 & V2) [6] are both large-scale action recognition benchmarks, including 108k and 220k action clips

Figure 1: Pouring [something]

Diving-48 [11] **i**s fine-grained action benchmark that is heavily dependent on temporal modeling containing 18k videos with 48 diving classes

Figure 1: ['Forward', '15som', 'NoTwis', 'PIKE']

FineGym [14] includes gymnastics action classes Gym288 and Gym99 that contain 288 and 99 action classes

Figure 1: Balance Beam

Ablation Studies on Something-V1

Table 4: **Ablation studies on SS-v1 dataset**. All models use TSN-ResNet50 [57] as the backbone. Top-1, top-5 accuracy (%), FLOPs (G) and paramaters (M) are shown.

(a) **Combinations of different kernels and context**. A single RSA layer is inserted into stage4.

kernel	context	FLOPs	params.	top-1	top-5
$\boldsymbol{\kappa}_n^{\mathrm{V}}$	$\boldsymbol{X}_n^{\mathrm{V}}$	32.3 G	23.4 M	44.8	73.8
$\boldsymbol{\kappa}_n^{\mathrm{R}}$	$\boldsymbol{X}_n^{\mathrm{V}}$	32.7 G	23.6 M	45.4	73.9
$\boldsymbol{\kappa}_n^{\mathrm{V}} + \boldsymbol{\kappa}_n^{\mathrm{R}}$	$\boldsymbol{X}_n^{\mathrm{V}}$	32.7 G	23.6 M	45.7	74.8
$\boldsymbol{\kappa}_n^{\mathrm{V}}$	$\boldsymbol{X}_n^{\mathrm{R}}$	32.7 G	23.4 M	46.2	75.4
$\boldsymbol{\kappa}_n^{\mathrm{R}}$	$\boldsymbol{X}_n^{\mathrm{R}}$	33.2 G	23.6 M	46.5	75.6
$\boldsymbol{\kappa}_n^{\mathrm{V}} + \boldsymbol{\kappa}_n^{\mathrm{R}}$	$\boldsymbol{X}_n^{\mathrm{R}}$	33.2 G	23.6 M	46.7	75.6
$\boldsymbol{\kappa}_n^{\mathrm{V}}$	$\boldsymbol{X}_n^{\mathrm{V}}+\boldsymbol{X}_n^{\mathrm{R}}$	32.7 G	23.4 M	46.5	75.6
$\boldsymbol{\kappa}_n^{\mathrm{R}}$	$\boldsymbol{X}_n^{\mathrm{V}}+\boldsymbol{X}_n^{\mathrm{R}}$	33.2 G	23.6 M	46.8	75.6
$\boldsymbol{\kappa}_n^{\mathrm{V}} + \boldsymbol{\kappa}_n^{\mathrm{R}}$	$\boldsymbol{X}_n^{\mathrm{V}}+\boldsymbol{X}_n^{\mathrm{R}}$	33.2 G	23.6 M	**47.0**	**75.7**

(b) **Latent dimension** D. Decomposing $\boldsymbol{H}$ significantly reduces the computation cost. OOM is an abbreviation of out-of-memory. 8 video clips per a single GPU machine are used.

D	FLOPs	params.	memory	top-1	top-5
-	62.9 G	32.0 M	OOM	OOM	OOM
8	32.2 G	20.5 M	8.8 GB	50.1	78.8
16	34.7 G	20.9 M	9.2 GB	51.3	78.8
32	39.6 G	21.7 M	10.2 GB	50.9	79.0
$C^{\mathrm{Q}}/2$	32.9 G	21.1 M	8.8 GB	51.1	79.1
C^{Q}	35.9 G	22.0 M	9.6 GB	**51.5**	**79.2**

(c) **Kernel size** M. In most cases, larger kernel results in the higher accuracy.

kernel size M	FLOPs	params.	top-1	top-5
$3 \times 3 \times 3$	28.5 G	20.3 M	49.4	77.6
$3 \times 5 \times 5$	30.2 G	20.7 M	50.5	78.7
$3 \times 7 \times 7$	32.6 G	21.2 M	50.7	78.9
$3 \times 9 \times 9$	35.8 G	22.0 M	51.1	79.1
$5 \times 7 \times 7$	35.9 G	22.0 M	**51.5**	**79.2**
$5 \times 9 \times 9$	41.3 G	23.3 M	51.2	78.9

(d) **Group** G. Hadamard product ($G = C^{\mathrm{Q}}$) performs the highest accuracy. Note that FLOPs are consistent with varying G due to the switched computation order.

# Groups G	FLOPs	params.	top-1	top-5
1	35.9 G	20.2 M	50.4	78.9
2	35.9 G	20.2 M	50.9	78.9
4	35.9 G	20.3 M	51.2	78.9
8	35.9 G	20.5 M	51.2	79.0
C^{Q}	35.9 G	22.0 M	**51.5**	**79.2**

Comparison to SOTA

State-of-the-art on three motion-centric video benchmarks

(a) **SS-V1&V2**. IN and IN21K and K400 denote ImageNet-1k, ImageNet-21K, and Kinetics-400 dataset, respectively. Our method achieves a new state-of-the-art accuracy on both datasets.

model	pre-trained	#frame	FLOPs ×clips	SS-V1 top-1	SS-V1 top-5	SS-V2 top-1	SS-V2 top-5
I3D [5] from [59]	IN	32	153 G×2	41.6	72.2	-	-
TSM-R50 [32]	IN	16	65 G×1	47.2	77.1	63.4	88.5
ir-CSN-152 [52]	-	32	97 G×10	49.3	-	-	-
SlowFast8×8-R50 [11]	K400	32	67 G×3	-	-	61.7	86.9
CT-Net-R50 [29]	IN	16	75 G × 1	52.5	80.9	64.5	89.3
STM-R50 [22]	IN	16	67 G×30	50.7	80.4	64.2	89.8
CorrNet-R101 [56]	-	32	187 G×10	50.9	-	-	-
TEA [30]	IN	16	70 G ×3	52.3	81.9	65.1	89.9
MSNet-TSM-R50 [24]	IN	16	67 G×1	52.1	82.3	64.7	89.4
NL-I3D [58] from [59]	IN	32	168 G×2	44.4	76.0	-	-
TimeSformer-HR [3]	IN	16	1703 G×3	-	-	62.2	-
TimeSformer-L [3]	IN	96	2380 G×3	-	-	62.4	-
ViViT-L [1]	IN21K & K400	32	N/A×4	-	-	65.4	89.8
RSANet-R50 (ours)	IN	8	36 G×1	51.9	79.6	64.8	89.1
RSANet-R50 (ours)	IN	16	72 G×1	54.0	81.1	66.0	89.8
RSANet-R50$_{EN}$ (ours)	IN	8+16	108 G×1	55.5	82.6	67.3	90.8
RSANet-R50$_{EN}$ (ours)	IN	8+16	108 G×2	**56.1**	**82.8**	**67.7**	**91.1**

(b) **Diving-48**. Top-1 accuracy, FLOPs are shown. Results in the upper compartment are from [3].

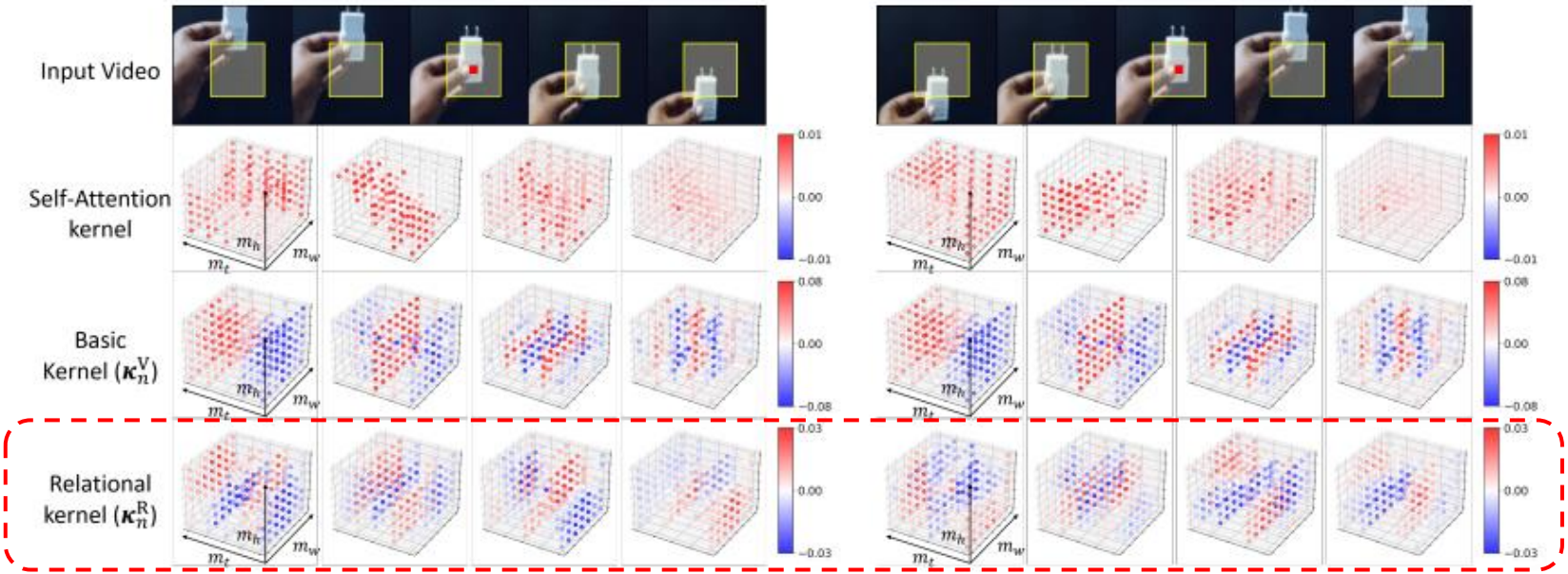
model	FLOPs ×clips	top-1
SlowFast-R101 [11]	213 G×3	77.6
TimeSformer [3]	196 G×3	75.0
TimeSformer-HR [3]	1703 G×3	78.0
TimeSformer-L [3]	2380 G×3	81.0
RSANet-R50	72G×2	**84.2**

(c) **FineGym**. The averaged per-class accuracy (%) is reported. All results in the upper compartment are from [41].

model	Gym288	Gym99
TRN [64]	33.1	68.7
I3D [5]	27.9	63.2
TSM [32]	34.8	70.6
TSM$_{\text{Two-stream}}$ [32]	46.5	81.2
RSANet-R50	**50.9**	**86.4**

Kernel visualization results on SS-V1

Reverse the temporal order of an input video clip(motion information)



(a) **'Moving something down.'** (origin) (b) **'Moving something up.'** (reversed order)

the **relational kernel** dynamically varies according to whether the object moves up or down but the **basic kernel** remains the same and the **self-attention** kernels are limited to aggregating the local context based on the query-key similarities.

Conclusion

1) Problems in self-attention:
- Individual query-key interaction for each kernel weight, resulting in output feature to be motion-agnostic.
- Limited expressive ability due to the softmax.
- Computational Complexity of Self-Attention.

2) Contributions:
- In-depth analysis on temporal modeling capabilities of recent dynamic feature transforms.
- The novel relational self-attention for capturing fine-grained temporal dynamics.
- State-of-the-art on three motion-centric video benchmarks.

3) Limitation:
- The computational efficiency of the RSA could be further improved.
- There will be a more generalized dynamic transform.
- Outperform recent works (Uniformer)

Reference

References

[1] A. Arnab et al., “ViViT: A Video Vision Transformer,” ICCV, 2021.
[2] G. Bertasius et al., “Is Space-Time Attention All You Need for Video Understanding?” ICML, 2021.
[3] R Child et al, "Generating long sequences with sparse transformers,” arXiv, 2019.
[4] A. Dosovitskiy et al., “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” ICLR, 2020.
[5] H Fan et al. "Multiscale vision transformers." ICCV, 2021.
[6] R. Goyal, et al. "The something something video database for learning and evaluating visual common sense,” ICCV, 2017.
[7] D Kondratyuk et al, "Movinets: Mobile video networks for efficient video recognition," CVPR, 2021.
[8] H Kwon et al., "MotionSqueeze: Neural Motion Feature Learning for Video Understanding," ECCV, 2020.
[9] H Kwon et al., "Learning self-similarity in space and time as generalized motion for video action recognition," ICCV, 2021.
[10] J Lee-Thorp et al. “FNet: Mixing Tokens with Fourier Transforms,” arXiv, 2021.
[11]1 Y. Li, et al, "RESOUND: Towards Action Recognition without Representation Bias," ECCV, 2018.
[12] X. Liu et al., "TSM: Temporal Shift Module for Efficient Video Understanding," ICCV, 2019.
[13] M. Ryoo et al., “TokenLearner: What can 8 Learned Tokens do for Images and Videos?” NeurIPS, 2021.
[14] D. Shao, et al. "Finegym: A hierarchical video dataset for fine-grained action understanding" CVPR, 2020.
[15] A. Srinivas et al., “Bottleneck Transformers for Visual Recognition,” CVPR, 2021.
[16] A. Vaswani et al. “Attention is All you Need.”, NeurIPS, 2017.
[17] Liu, Ze et al. “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows,” ICCV, 2021.
[18] Liu, Ze et al. “Video Swin Transformer,” arXiv, 2021.
[19] M. Kim et al., “Relational Self-Attention: What’s Missing in Attention for Video Understanding,” NeurIPS, 2021.

https://cvlab.postech.ac.kr/research/RSA/
https://www.eiric.or.kr/community/webinar_detail.php?Seq=71&totalCnt=67

Thanks

Any Questions?