

# GTA: Global Temporal Attention for Video Action Understanding

Bo He<sup>\*1</sup>  
bohe@umd.edu

Xitong Yang<sup>\*1</sup>  
xyang35@cs.umd.edu

Zuxuan Wu<sup>2</sup>  
zxwu@fudan.edu.cn

Hao Chen<sup>1</sup>  
chenh@umd.umd

Ser-Nam Lim<sup>3</sup>  
sernamlim@fb.com

Abhinav Shrivastava<sup>1</sup>  
abhinav@cs.umd.edu

<sup>1</sup> University of Maryland,  
College Park, MD, USA

<sup>2</sup> Fudan University,  
Shanghai, China

<sup>3</sup> Facebook AI,  
Sunnyvale, CA, USA

## Abstract

Self-attention learns pairwise interactions to model long-range dependencies, yielding great improvements for video action recognition. In this paper, we seek a deeper understanding of self-attention for temporal modeling in videos. We first demonstrate that the entangled modeling of spatio-temporal information by flattening all pixels is sub-optimal, failing to capture temporal relationships among frames explicitly. To this end, we introduce Global Temporal Attention (GTA), which performs global temporal attention on top of spatial attention in a decoupled manner. We apply GTA on both pixels and semantically similar regions to capture temporal relationships at different levels of spatial granularity. Unlike conventional self-attention that computes an instance-specific attention matrix, GTA directly learns a global attention matrix that is intended to encode temporal structures that generalize across different samples. We further augment GTA with a cross-channel multi-head fashion to exploit channel interactions for better temporal modeling. Extensive experiments on 2D and 3D networks demonstrate that our approach consistently enhances temporal modeling and provides state-of-the-art performance on three video action recognition datasets.

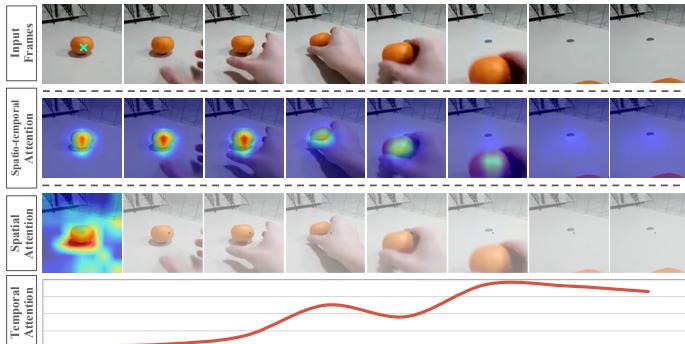
## 1 Introduction

Attention mechanisms have demonstrated impressive achievements in a wide range of tasks such as language modeling [1, 2], speech recognition [3] and image classification [4, 5]. One of the most effective attention methods is self-attention, which learns self-alignment via dot product operations, computing pairwise similarities between a pixel (*i.e.*, query) and

© 2021. The copyright of this document resides with its authors.

It may be distributed unchanged freely in print or electronic forms.

\*indicates equal contribution.



**Figure 1: Top:** input frames (action: *removing something to reveal something*). The green cross-mark indicates the query position. **Center:** spatio-temporal attention generated by NL blocks. The attention is biased towards the appearance similarity, which fades overtime ignoring temporal clues; thus, the model generates incorrect prediction: *putting something in front of something*. **Bottom:** the decoupled NL blocks generate spatial attention maps within the query frame and temporal attention weights across different time steps. The temporal attention has larger values at the key frames that are critical for recognizing the action (*i.e.*, revealing something), and the model gives the correct prediction. GTA is built upon the decoupled framework and advances the temporal attention to a more effective design.

other pixels (*i.e.*, key) to modulate the transformed inputs (*i.e.*, value). For action recognition [24], this requires: (i) flattening all pixels in a video, regardless of their spatial and temporal locations, into a huge vector; (ii) sharing the same set of parameters for all pixels to derive the query/key/value; and (iii) generating a joint attention map for both spatial and temporal context.

In this paper, we seek a better understanding of self-attention for temporal modeling in videos. In particular, we wish to answer the following questions: (i) Is treating all pixels in space and time as a flattened vector to perform dot-product sufficient for temporal modeling? (ii) Is dot product based self-attention really necessary for capturing temporal relationships across different frames?

In contrast to the conventional use of self-attention for video recognition, we posit that temporal attention should be *disentangled* from spatial attention, since they focus on different aspects. As shown in Figure 1, the spatial attention tends to capture appearance similarity (*i.e.*, the orange), while the temporal attention is more focused on frames that are important for recognizing the action (*i.e.*, revealing something). When these two types of attention are modeled together (Figure 1 Center), the attention is biased towards the appearance similarity, dominating any temporal context.

In addition, we argue that dot product based self-attention is *not even suitable* for temporal modeling. Standard self-attention produces instance-specific attention weights, conditioned on pairwise interactions. In the spatial domain, it can attend to salient regions for improved performance. When used for temporal modeling, it ignores the ordering of frames as self-attention is known to be permutation invariant [8]. For instance, if we shuffle two pixels temporally, their relationship will be the same, producing the same output. This is not sufficient for differentiating actions like “reveal something” and “cover something”. We hypothesize that temporal modeling requires learning a *global* temporal structure that gener-

alizes across different samples rather than relying on pairwise interactions across time steps.

In light of this, we introduce Global Temporal Attention (GTA), for video action recognition. In particular, we first decouple the traditional spatio-temporal self-attention into two successive steps—a standard self-attention in the spatial domain within each frame followed by the proposed GTA module to capture temporal relationships across different frames. Moreover, we not only apply GTA to each pixel location along the temporal dimension but also “superpixels”—pixels in a region share similar semantic meanings. This enables our model to capture temporal relationships at different levels of spatial granularity. Unlike computing pairwise frame interactions with dot product, GTA directly learns a global attention matrix that is randomly initialized to be instance-independent. The intuition of the global attention matrix is to not rely on pairwise frame relations without specific ordering information or individual sample information, but to learn a global task-specific weight matrix considering temporal structures that generalize across different samples. To exploit information across different channels, we split feature maps into multiple groups along the channel-dimension, and for each group we apply GTA in a multi-head fashion such that each head focuses on different aspects of the inputs. Then, outputs from different channel groups are further aggregated to produce a unified representation.

We conduct extensive experiments on Something-Something [15] and Kinetics-400 [29]. Our proposed GTA outperforms the traditional spatio-temporal self-attention by clear margins, and achieves state-of-the-art results on these three datasets. We also provide a side-by-side comparison with recent NL variants [8, 9, 46] to show the superior performance of GTA in temporal modeling. We summarize our main contributions as follows. First, we provide an in-depth analysis of the sub-optimal design of the spatio-temporal self-attention and propose to decouple attention across the two dimensions. Second, we introduce GTA, which improves the conventional temporal attention by introducing: (i) temporal modeling at both pixel and region levels; (ii) a global attention matrix for all samples; (iii) a cross-channel multi-head design for incorporating channel interactions.

## 2 Related Work

**Temporal Modeling in Action Recognition.** A large family of research in action recognition focuses on the effective modeling of temporal information in videos. Early work simply aggregates the frame/clip-level features across time via average pooling [28, 42] or feature encoding like ActionVLAD [12], without considering the temporal relationships of video frames. Later on, two-stream networks [35], 3D convolution networks (CNNs) [26, 38] and recurrent neural networks (RNNs) [9, 47] are used to model the spatial and temporal context in videos. Recently, various temporal modules are proposed to capture temporal relations, such as TRN [51] based on relation networks, Timeception [24] based on multi-scale temporal convolutions, and SlowFast [1] based on slow and fast branches capturing spatial and motion information, respectively. TSM [3] adopts a channel shifting operation along the time dimension to enable temporal modeling on 2D CNN networks. STM [27], TEA [31] and MSNet [30] encode the motion information into the network by extracting motion features between adjacent frames.

**Non-Local and Self-Attention.** Modeling long-range relations in feature representations has a long history [8, 10, 17, 19, 33, 37] and has proven to be effective in various tasks, such as machine translation [40], generative modeling [38], image recognition [8, 21, 42], object detection [8, 20, 42], semantic segmentation [8, 24, 49] and visual question answering [32]. In

computer vision, Non-local Network (NL) [42] is proposed to model the pixel-level pairwise similarities to encode long-range dependencies. SENet [29] uses a Squeeze-and-Excitation block to model inter-dependencies along the channel dimension. GCNet [2], CGNL [46] and DANet [13] further improve the vanilla NL by integrating pixel-wise and channel-wise attention. CCNet [23] improves the efficiency of NL by computing the contextual information of the pixels on its crisscross path instead of the global region. GloRe [8] proposes the relation reasoning via graph convolution on a region-based graph in the interaction space to capture the global information.

In this work, we present a novel way to model temporal relationships and bring new perspectives for a better understanding of the attention mechanism utilized in video action recognition. Our approach learns global temporal attention that generalizes well across different samples as opposed to using pairwise interactions with dot product in self-attention.

## 3 Approach

### 3.1 Background

Extending the self-attention module [40] for language tasks, the non-local block (NL) [42] takes as input flattened pixels in spacetime to model pairwise interactions, as shown in Figure 2(a). More formally, given an input feature map  $X \in \mathbb{R}^{N \times C}$ , three linear projections are applied to obtain key ( $K$ ), query ( $Q$ ), and value ( $V$ ) representations, where  $C$  is the channel dimension of the feature map. We use  $N = THW$  to denote the total number of positions in both space and time dimensions, where  $T$ ,  $H$  and  $W$  are the number of time steps, height and width of the feature map, respectively. The three projections can be written as  $Q = XW_Q$ ,  $K = XW_K$ ,  $V = XW_V$ , parameterized by three weight matrices  $W_Q, W_K, W_V \in \mathbb{R}^{C \times C}$  respectively. The output of the self-attention operation is computed as a weighted sum of the value representations. Here, the weight is defined by the attention weight matrix  $M \in \mathbb{R}^{N \times N}$ , where each element denotes a scaled dot product between the query pixel and the corresponding key pixel, followed by a softmax normalization:

$$A = MV, \quad M = \text{softmax}\left(\frac{QK^T}{\sqrt{C}}\right). \quad (1)$$

The attention output is incorporated into the backbone network via a final linear projection  $W^O \in \mathbb{R}^{C \times C}$  and a residual connection [18]:

$$Y = X + AW^O, \quad (2)$$

An optional normalization layer (e.g., BatchNorm [25] and LayerNorm [11]) can be used before the residual connection, and we drop it here for clarity.

### 3.2 Decoupled Spatial and Temporal Self-Attention

Although self-attention has been widely used in action recognition for capturing spatio-temporal dependencies, we argue in this paper that the coupled modeling of spatial and temporal self-attention prevents the model from learning effective temporal attention. First, when sharing the same transformation matrices for key, query and value, it fails to differentiate between spatial and temporal contexts. This is unsatisfactory for temporal modeling as

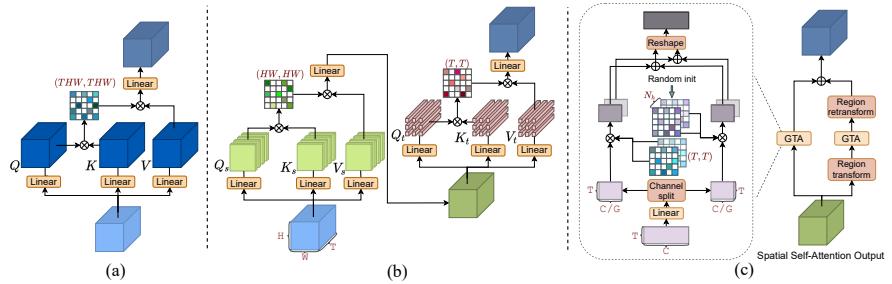


Figure 2: (a) **Standard self-attention for action recognition**, which computes pairwise similarities between a pixel (query) with other pixels (key) in the spacetime domain. (b) **Decoupled spatial and temporal self-attention**, which uses separated key/query/value representations for spatial and temporal attention and aggregates spatial and temporal context in a separate manner. (c) **Global temporal attention**, which learns two randomly initialized global attention maps at the pixel-level and the region-level, respectively. Regions are derived automatically with a learned transformation matrix. Inside the rectangular (dashed line), the spatial dimension of feature maps is omitted. GTA is also applied in a cross-channel multi-head fashion, where feature maps are split along the channel dimension into G groups (only 2 groups are shown for simplicity). Residual connections are omitted here for simplicity. See texts for more details.

we need to consider temporal structures of videos instead of simply computing the salient regions by performing self-attention in the spatial domain. Moreover, when the two attentions are modeled and aggregated jointly, the combined attention tends to be biased towards the appearance similarity as the temporal attention is dominated by the spatial one (see Figure 1). Based on this observation, we propose the decoupled spatial and temporal self-attention in Figure 2 (b), which breaks down the standard self-attention block into a spatial self-attention block followed by a temporal self-attention block. We will provide a more in-depth analysis of the decoupled self-attention design via experiments in Section 4.1.

Formally, given the input feature map  $X$ , we first obtain the three projections through:  $Q_s = XW_Q^s, K_s = XW_K^s, V_s = XW_V^s$ , where the subscript/superscript  $s$  is used to differentiate from the functions in temporal attention. We then perform the space-only attention on all spatial positions within each frame:

$$A_s(t) = \text{softmax} \left( \frac{Q_s(t)K_s(t)^T}{\sqrt{C}} \right) V_s(t), \quad Y_s(t) = X(t) + A_s(t)W_O^s, \quad (3)$$

where  $t \in \{1, \dots, T\}$  denotes different time steps. The output of the spatial attention block is then used as input to the temporal attention block, which performs similar attention operations yet in time only:

$$A_t(i, j) = \text{softmax} \left( \frac{Q_t(i, j)K_t(i, j)^T}{\sqrt{C}} \right) V_t(i, j), \quad Y_t(i, j) = Y_s(i, j) + A_t(i, j)W_O^t, \quad (4)$$

where  $i \in \{1, \dots, H\}, j \in \{1, \dots, W\}$  denote different spatial positions, and  $Q_t = Y_s W_Q^t, K_t = Y_s W_K^t, V_t = Y_s W_V^t$ . Note that although the idea of processing spatial and temporal information separately has been explored before for video understanding [36, 33], the effect of disentangling the two dimensions in self-attention is *unknown* in prior work.

### 3.3 Global Temporal Attention

We now introduce GTA, which is built upon the decoupled self-attention framework and advances the temporal attention to a more effective design. GTA aims to learn a global attention map that considers temporal structures and generalizes well for all samples.

Formally, given the input feature map  $X \in \mathbb{R}^{T \times HW \times C}$  generated by the spatial self-attention block, GTA models temporal relationships at two different levels of spatial granularity: *pixel-level* and *region-level*. For Pixel GTA, all positions in the spatial domain (*i.e.*,  $HW$ ) are treated individually as different samples and temporal modeling is performed along the time axis  $T$ . As for Region GTA, we first project the spatial domain to  $K$  semantic regions at each time step  $t$ . This is achieved by grouping similar pixels with related semantic meanings into the same region [■]:  $X_G(t) = G_R(t)X(t)$ , where the region transformation matrix  $G_R(t) = W_G X(t)^T$  and  $W_G \in \mathbb{R}^{K \times C}$  is a learnable weight matrix. Then, temporal modeling is performed across frames on each region individually in the same manner as Pixel GTA, followed by a transposed region transformation matrix  $G_R^T$  to reproduce the pixel-level spatial domain. Similar to Eqn. 2, the output of GTA can be written as:

$$Y = X + \underbrace{A_P W_P^O}_{\text{Pixel GTA}} + \underbrace{A_R W_R^O}_{\text{Region GTA}}. \quad (5)$$

Unlike conventional self-attention where the attention map is produced by pairwise dot-product interactions (Eqn. 1), we train attention maps that do *not* depend on individual pixel relationships. In particular, we directly learn randomly initialized weight matrices  $\hat{M}_P, \hat{M}_R \in \mathbb{R}^{T \times T}$  to modulate the value representation of Pixel and Region GTA, respectively:

$$A_P = \hat{M}_P V_P, \quad A_R = G_R^T (\hat{M}_R V_R), \quad (6)$$

The idea of using a learned global attention matrix rather than pairwise dot product is that pairwise interactions fluctuate across different samples, lacking a global temporal consistency at the dataset level. In addition, the standard self-attention fails to consider the ordering of sequences [■]—if we shuffle the pixels used to compute the attention map (*i.e.* Eqn. 1), the attention value between a pair would still be the same in the matrix, thus the output will not change, which is not what we desire.

**Cross-channel Multi-head GTA.** The attention matrix  $\hat{M}$  in Eqn. 6 is used to learn a linear combination of  $V \in \mathbb{R}^{T \times C}$ <sup>1</sup> across different time steps, without considering feature interactions in the channel dimension. We further improve temporal modeling by incorporating channel interactions. We split  $C$  into  $G$  groups, and for each group and we apply a multi-head GTA. In particular, for the  $g$ -th group, the outputs of the multi-head attention  $\text{MH}_g$  is:

$$\text{MH}_g = \text{Concat}_{k=1}^{N_h} (\hat{M}_g^k V_g) \in \mathbb{R}^{N_h \times T \times \lfloor \frac{C}{G} \rfloor}, \quad (7)$$

where  $\hat{M}_g^k \in \mathbb{R}^{T \times T}$  represents the  $k$ -th head for the  $g$ -th group,  $V_g \in \mathbb{R}^{T \times \lfloor \frac{C}{G} \rfloor}$  denotes the value for the  $g$ -th group and  $N_h$  denotes the number of heads used. Each head focuses on distinct temporal attention patterns. To capture interactions across different groups, we sum the outputs along the channel dimension between different groups to produce  $\text{MH}_G$  as:

<sup>1</sup>We omit the subscripts  $P$  and  $R$  for  $A$ ,  $\hat{M}$  and  $V$ , as the same operations are applied to both Pixel and Region GTA.  $HW$  are considered as different samples and we omit it for brevity.

Model	FLOPs	#Params	SSv1	SSv2
R2D-50	32.7 G	23.9 M	17.0	26.8
+ NL	61.1 G	31.2 M	31.2	50.7
+ DNL	49.9 G	31.2 M	38.8	55.5
+ GTA	50.2 G	31.2 M	<b>50.6</b>	<b>63.5</b>
SlowFast-R50	131.4 G	34.0 M	50.9	63.4
+ NL	239.9 G	41.4 M	51.7	63.9
+ DNL	169.1 G	41.4 M	52.0	64.1
+ GTA	169.9 G	41.4 M	<b>53.4</b>	<b>64.9</b>

Table 1: Compare GTA with the standard / decoupled non-local block (NL / DNL).

Method	GFLOPs×views	Top-1	Top-5
TSM [16]	86×30	74.7	91.4
bLVNet-TAM [10]	93×9	73.5	91.2
MSNet [20]	87×10	76.4	-
S3D-G [20]	143×N/A	77.2	93.0
I3D+NL [10]	359×30	77.7	93.3
CorrNet-R101 [10]	224×30	79.2	-
R2D-R50 + NL	77×30	74.8	91.5
R2D-R50 + GTA	62×30	75.9	92.2
SlowFast-R101 + NL	137×30	78.9	93.9
<b>SlowFast-R101 + GTA</b>	137×30	<b>79.8</b>	<b>94.1</b>

Table 2: Comparisons with state-of-the-art methods on Kinetics-400 dataset.

$$\text{MH}_G = \sum_{g=1}^G \text{MH}_g \in \mathbb{R}^{N_h \times T \times \lfloor \frac{C}{G} \rfloor}, \quad (8)$$

which mixes information across channels in different groups. In order for  $\text{MH}_G$  to have the same size as  $X \in \mathbb{R}^{T \times C}$  for residual addition, one can transform  $\text{MH}_G$  with an additional layer. Instead, we simply set  $N_h$  to be  $G$  and reshape  $\text{MH}_G$  to be the same size of  $\mathbb{R}^{T \times C}$ .

## 4 Experiments

We extensively evaluate our approach on three video action benchmarks, including two temporal-related datasets: Something-Something (v1&v2) [16], and a large-scale dataset that is less sensitive to temporal relationships: Kinetics-400 (K400) [8]. As we aim to improve temporal modeling for video action recognition, our experiments focus more on temporal sensitive datasets (SSv1 and SSv2). GTA is flexible and can be easily inserted into existing 2D and 3D backbones. In our experiments, we adopt the standard R2D-50 network [18] and the SlowFast-R50 network [10] as our 2D/3D backbones. More dataset-specific training and testing details are available in the supplementary material.

### 4.1 Main Results

**Effectiveness of GTA in a decoupled framework.** We report the results of GTA using both 2D and 3D backbones and compare with the alternative approaches: (1) standard non-local block (NL) [20], which is a variant of self-attention that flattens all pixels in space and time dimension into a huge vector; (2) decoupled non-local block (DNL), which breaks down NL into spatial self-attention followed by temporal self-attention. For both of our approaches and the compared baselines, we apply five blocks (2 to  $\text{res}_3$  and 3 to  $\text{res}_4$  for every other residual block) in the backbone networks unless specified, following [10].

Table 1 summarizes the comparison results. We first observe a huge gap between the performance of 2D and 3D backbones, which shows the importance of utilizing temporal information for SSv1&SSv2 datasets. Notably, we see that by simply separating temporal self-attention from spatial self-attention, DNL outperforms NL on both backbones, while requiring 20%-30% less computation cost. Compared to NL, DNL offers a 7.6% / 4.8% gain on SSv1 / SSv2 in the 2D setting. This suggests that the spatial and temporal self-attentions should be treated *separately* to capture more informative temporal contexts. Finally, GTA

Method	Backbone	Pretrain	Frames $\times$ Crops $\times$ Clips	SSv1		SSv2	
				Top-1	Top-5	Top-1	Top-5
TRN [30]	BNInception	ImgNet	8 $\times$ 1 $\times$ 1	34.4	-	48.8	-
TSM [31]	2D R50	ImgNet	8 $\times$ 1 $\times$ 1	45.6	74.2	58.8	85.4
TSM [31]	2D R50	ImgNet	16 $\times$ 1 $\times$ 1	47.3	77.1	61.2	86.9
TSM <sub>RGB+Flow</sub> [32]	2D R50	ImgNet	(16+16) $\times$ 1 $\times$ 1	52.6	81.9	65.0	89.4
MSNet [33]	2D R50+TSM	ImgNet	8 $\times$ 1 $\times$ 1	50.9	80.3	63.0	88.4
MSNet [33]	2D R50+TSM	ImgNet	16 $\times$ 1 $\times$ 1	52.1	82.3	64.7	89.4
MSNet <sub>En</sub> [33]	2D R50+TSM	ImgNet	(16+8) $\times$ 1 $\times$ 10	<u>55.1</u>	<b>84.0</b>	<u>67.1</u>	<b>91.0</b>
ECO [34]	3D R18+BNInc	K400	16 $\times$ 1 $\times$ 1	41.4	-	-	-
ECO <sub>En</sub> Lite [34]	BNInc+3D R18	K400	92 $\times$ 1 $\times$ 1	46.4	-	-	-
I3D+NL [35]	3D R50	K400	32 $\times$ 3 $\times$ 2	44.4	76.0	-	-
I3D+NL+GCN [35]	3D R50	K400	32 $\times$ 3 $\times$ 2	46.1	76.8	-	-
S3D-G [36]	3D Inception	ImgNet	64 $\times$ 1 $\times$ 1	48.2	78.7	-	-
CorrNet [37]	3D CorrNet-50	-	32 $\times$ 1 $\times$ 10	48.5	-	-	-
CorrNet [37]	3D CorrNet-101	-	32 $\times$ 3 $\times$ 10	51.1	-	-	-
TEA [38]	3D R50	ImgNet	8 $\times$ 1 $\times$ 1	48.9	78.1	-	-
TEA [38]	3D R50	ImgNet	16 $\times$ 3 $\times$ 10	52.3	81.9	65.1	89.9
GTA	2D R50	ImgNet	8 $\times$ 1 $\times$ 1	50.6	78.8	63.5	88.6
GTA	2D R50	ImgNet	16 $\times$ 1 $\times$ 1	52.0	80.5	64.7	89.3
GTA	2D R50+TSM	ImgNet	8 $\times$ 1 $\times$ 1	51.6	79.8	63.7	88.9
GTA	2D R50+TSM	ImgNet	16 $\times$ 1 $\times$ 1	53.7	81.7	65.3	89.6
GTA <sub>En</sub>	2D R50+TSM	ImgNet	(16+8) $\times$ 3 $\times$ 2	<b>56.5</b>	<u>83.1</u>	<b>68.1</b>	<b>91.1</b>

Table 3: Comparisons with state-of-the-art methods on Something-Something v1 & v2 datasets. Top-1 and Top-5 accuracy on validation set are reported here. **Bold** and underline shows the highest and second highest results.

produces the best results on the two datasets with both 2D and 3D backbones with reduced FLOPs comparing to NL. For example, on the 2D backbone, GTA further outperforms DNL by 11.8% / 8.0% on SSv1, SSv2, respectively, confirming the effectiveness of GTA for temporal modeling. On a 3D backbone, we observe similar trends with gains. This highlights the compatibility of GTA for both 2D and 3D networks. It is also noteworthy that 2D networks can achieve comparable performance with 3D backbones when equipped with GTA.

## 4.2 Comparison with State-of-the-art

**Kinetics-400** Table 2 presents the comparative results with other state-of-the-art methods on Kinetics-400. The first section of the table shows the methods based on 2D CNN network. The second section contains the models with 3D CNN backbone. The third section illustrates the comparison of our GTA and NL added to 2D and 3D CNN backbones. We can see that GTA achieves consistent improvement over the NL counterpart on 2D and 3D CNN backbones. And adding GTA to SlowFast-R101 can achieve 79.8% top-1 accuracy on Kinetics-400 dataset, which is the state-of-the-art performance on Kinetics-400.

**Something-Something v1&v2** We also compare our approach with the state-of-the-art methods on SSv1 & SSv2 datasets. As shown in Table 7, given 8 input frames, our approach based on 2D RestNet-50 with TSM backbone achieves 51.6% and 63.7% on SSv1 and SSv2 at top-1 accuracy, respectively. Specifically, with the same number of input frames, our approach outperforms TRN [30] which utilizes relation networks, and MSNet [33] which incorporates the motion features. This demonstrates that our proposed GTA is more effective in temporal modeling. Our approach also achieves superior results when compared with the recent work that leverages additional modules to improve 3D CNN backbones, such as the non-local block (I3D+NL [35]), GCN (I3D+NL+GCN [35]), the correlation operation

Pixel	Region	CCMH	Top-1	$\Delta$
✓	✓	✓	50.6	-
✓		✓	49.6	-1.0
	✓	✓	47.9	-2.7
✓	✓		49.4	-1.2
✓			49.1	-1.5
	✓		46.3	-4.3

Table 4: Contribution of different components in GTA.

Model	Original	Decoupled
GTA	-	<b>50.6</b>
NL [2]	31.2	38.8
CGNL [3]	26.7	37.4
GCNet [4]	28.4	39.0
GloRe [5]	33.2	38.6

Table 5: Comparisons with recent NL variants.

Model	FLOPs	#Params	Top-1	$\Delta$
R2D-50	32.7 G	23.9 M	17.0	-
+SA	41.7 G	27.5 M	17.9	+0.9
+TA	41.0 G	27.5 M	37.6	+20.6
+SA+TA	49.9 G	31.2 M	38.8	+21.8
+SA+TAPE	49.9 G	31.2 M	48.4	+31.4
+SA+GTA	50.2 G	31.2 M	50.6	+33.6

Table 6: Impact of SA, TA and temporal order.

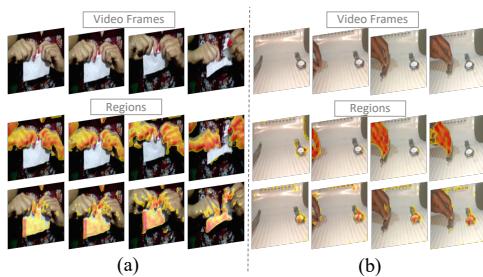
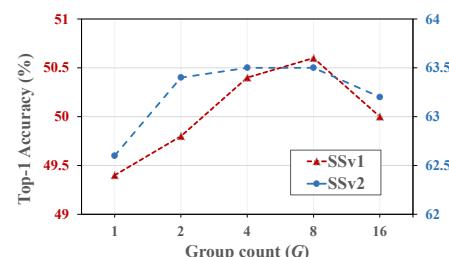


Figure 3: Regions visualization: (a) “Tearing” Figure 4: Impact of group count. Top-smth. into two pieces”; (b) “Moving” smth. 1 closer to smth.”. The second and third rows datasets are reported here. are regions obtained by Region GTA.



(CorrNet [2]), and the multiple temporal aggregation module and the motion excitation module (TEA [3]). Finally, we evaluate the ensemble model ( $GTA_E$ ) by averaging output prediction scores of the 8-frame and 16-frame models, and obtain 56.5% and 68.1% at top-1 accuracy on SSv1 and SSv2, respectively, which achieves the state-of-the-art performance.

### 4.3 Ablative Studies

We conduct extensive ablation studies on SSv1 using R2D backbone. More analysis are available in the supplementary material.

**Contribution of Different Components.** We first validate the contribution of each component in GTA by removing them from the full model. As shown in Table 4, while Pixel GTA plays a more important role than Region GTA, the combination of these two modules yields the best result, achieving more than 1% improvement compared to using each of them alone. It indicates that Pixel GTA and Region GTA are *complementary* to each other, focusing on learning temporal relationships at different levels of spatial granularity. We further visualize regions that are automatically discovered by Region GTA in Figure 3. We can see that Region GTA is capable of discovering regions that share similar semantic meanings. For example, in the first video, the “hand” and the “paper” are automatically identified as different regions, while the “hand” and the “watch” are detected in the second video. Table 4 also shows the contribution of the cross-channel multi-head (CCMH) design when the group size is set to 8. Specifically, CCMH has a larger impact on Region GTA than Pixel GTA (1% gain v.s. 0.5% gain) and we hypothesize that modeling temporal relationships at the region level

is more challenging and requires channel interactions. With the improved performance of Region GTA, the fusion of pixel-level and region-level information becomes more beneficial when CCMH is applied (1.0% gain v.s. 0.3% gain w/o CCMH).

**Temporal modeling in NL variants.** Recent work has focused on improving the vanilla non-local block by introducing channel-wise attention [4, 43] or graph-based reasoning [4]. Although these variants have been applied to video action recognition, their capacity to model temporal relations is relatively underexplored. In Table 5, we provide a side-by-side comparison with these NL variants and their decoupled version on SSv1. We first observe that all three variants fail to achieve satisfying improvements over the vanilla NL (31.2%). In particular, the use of extra channel-wise attention ( CGNL [43], GCNet [4]) leads to even worse results, indicating that the entangled modeling of spatial, temporal and channel interactions in fact hinders the learning of temporal relationships. Interestingly, by simply decoupling the spatial and temporal operations, substantial improvements can be achieved for all three variants and the results are comparable with DNL (38.8%). Nevertheless, our GTA outperforms these NL variants by clear margins, which demonstrates its superior capacity to model temporal information.

**Miscellaneous.** In Table 6, we compare the contribution of spatial and temporal self-attention modules, as well as the impact of modeling temporal order in temporal self-attention. As the SSv1 dataset relies highly on temporal relationships, applying spatial self-attention (SA) alone in the spatial domain slightly improves the backbone network (0.9% gain). In contrast, using the temporal self-attention (TA) provides much more significant improvements (20.6% gain). Adding positional encoding to the temporal self-attention module (TAPE) further improves the performance by 9.6%, which proves the importance of modeling temporal order information. Finally, our GTA achieves the best result with a negligible increase in computation cost. It is worth noting that our Pixel GTA (without applying GTA to regions) already outperforms TAPE no matter whether CCMH is used or not (49.1% / 49.6% in Table 4). This verifies that our GTA design is more effective in temporal modeling than temporal self-attention and positional encoding.

We also evaluate different values of group count used in GTA in Figure 4. We can see that using a group count larger than 1 can largely improve the performance, which demonstrates the importance of channel interactions in GTA. And a group count of 8 offers the best performance on SSv1 and SSv2. When the group count becomes larger than 8, the performance drops because the number of channels in each group becomes too small.

## 5 Conclusion

In this paper, we present Global Temporal Attention (GTA), which is designed for improved temporal modeling in video tasks. GTA is built upon a decoupled self-attention framework, where temporal attention is disentangled from the spatial attention to prevent being dominated by the spatial one. We apply GTA to model the temporal relationships at both pixel-level and region-level. Moreover, GTA directly learns a global, instance-independent attention matrix that generalizes well across different samples. A cross-channel multi-head mechanism is also designed to further improve the temporal modeling in GTA. Experimental results demonstrate that our proposed GTA effectively enhances temporal modeling and achieves state-of-the-art results on three challenging video action benchmarks.

**Acknowledgements.** This work was supported by the Air Force (STTR awards FA865019P6014, FA864920C0010), DARPA SemaFor program (HR001120C0124), and an independent gift from Facebook AI.

## References

- [1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- [2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- [3] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *CVPR*, 2005.
- [4] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *ICCVW*, 2019.
- [5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.
- [6] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *CVPR*, 2019.
- [7] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In *NeurIPS*, 2015.
- [8] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. In *ICLR*, 2020.
- [9] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Trevor Darrell, and Kate Saenko. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.
- [10] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *ICCV*, 1999.
- [11] Quanfu Fan, Chun-Fu Richard Chen, Hilde Kuehne, Marco Pistoia, and David Cox. More is less: Learning efficient video representations by big-little network and depth-wise temporal aggregation. In *NeurIPS*, 2019.
- [12] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019.
- [13] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, 2019.
- [14] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. Action-vlad: Learning spatio-temporal aggregation for action classification. In *CVPR*, 2017.
- [15] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

- [16] 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.
- [17] Saurabh Gupta, Bharath Hariharan, and Jitendra Malik. Exploring person context and local scene context for object detection. *arXiv preprint arXiv:1511.08177*, 2015.
- [18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [19] Jeremy Heitz and Daphne Koller. Learning spatial context: Using stuff to find things. In *ECCV*, 2008.
- [20] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *CVPR*, 2018.
- [21] Jie Hu, L Longfei Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convnets. In *NeurIPS*, 2018.
- [22] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.
- [23] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *ICCV*, 2019.
- [24] Noureddien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Timeception for complex action recognition. In *CVPR*, 2019.
- [25] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [26] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *PAMI*, 2012.
- [27] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *ICCV*, 2019.
- [28] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.
- [29] 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.
- [30] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. Motionsqueeze: Neural motion feature learning for video understanding. In *ECCV*, 2020.
- [31] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for action recognition. In *CVPR*, 2020.
- [32] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, 2019.

- [33] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014.
- [34] Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter W. Battaglia, and Timothy P. Lillicrap. A simple neural network module for relational reasoning. In *NeurIPS*, 2017.
- [35] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, 2014.
- [36] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *AAAI*, 2017.
- [37] Thomas M Strat and Martin A Fischler. Context-based vision: recognizing objects using information from both 2 d and 3 d imagery. *PAMI*, 1991.
- [38] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.
- [39] 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.
- [40] 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*, 2017.
- [41] Heng Wang, Du Tran, Lorenzo Torresani, and Matt Feiszli. Video modeling with correlation networks. In *CVPR*, 2020.
- [42] 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.
- [43] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, 2018.
- [44] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2017.
- [45] 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.
- [46] Kaiyu Yue, Ming Sun, Yuchen Yuan, Feng Zhou, Errui Ding, and Fuxin Xu. Compact generalized non-local network. In *NeurIPS*, 2018.
- [47] 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*, 2015.

- 
- [48] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.
  - [49] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *CVPR*, 2018.
  - [50] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, 2018.
  - [51] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. Eco: Efficient convolutional network for online video understanding. In *ECCV*, 2018.

## Appendix

Section A reports additional results on the test set of Something-Something v1&v2. Section B presents more ablative study results of GTA. Section C elaborates on GTA that is designed for better temporal modeling. Section D shows more visualization results of global temporal attention weights, transformed regions and swapped attention. Finally, Section E provides dataset-specific implementation details on training and testing.

## A Testing Results on Something v1&v2

We compare the performance of our approach on the test set with the state-of-the-art methods on Something-Something v1 & v2 datasets. As is shown in Table 7, our approach based on 2D RestNet-50 with TSM backbone achieves 49.8% and 66.9% on SSv1 and SSv2 at top-1 accuracy, respectively. Although on SSv1 dataset, it is still below the  $TSM_{RGB+Flow}$ ,  $TSM_{RGB+Flow}$  is based on the two-stream network and utilizes additional optical flow information. With only RGB input, our GTA achieves the best performance among the recently proposed STM [27] and bLVNet-TAM [10] on 2D CNN backbone; I3D+NL+GCN [43] and TEA [31] on 3D CNN backbone.

Method	Backbone	Frames	SSv1	SSv2
TRN <sub>RGB+Flow</sub> [30]	BNInc	8+8	40.7	56.2
TSM [30]	2D R50	16	46.0	64.3
$TSM_{RGB+Flow}$ [30]	2D R50	16+16	<b>50.7</b>	<b>66.6</b>
STM [27]	2D R50	16	43.1	63.5
bLVNet-TAM [10]	2D R101	64	48.9	-
ECO <sub>En</sub> Lite [30]	BNInc+3D R18	92	42.3	-
I3D+NL+GCN [43]	3D R50	32	45.0	-
TEA [31]	3D R50	16	46.6	63.2
<b>GTA<sub>En</sub></b>	2D R50+TSM	16+8	<b>49.8</b>	<b>66.9</b>

Table 7: Results on the test set of Something-Something v1 & v2 datasets.

## B More Ablative Studies

**Impact of inserting positions and number of blocks** Table 8 explores the performance of different inserting positions and the number of blocks inserted. We see that even a single GTA block inserted at  $res_3$  or  $res_4$  can bring significant improvement over the baseline. However, the enhancement on  $res_5$  is relatively minor. We hypothesize that the final residual stage loses too much fine-grained spatial information, which may hinder the learning of temporal attention at the pixel-level and the region-level. Following the common practice [44], our full model inserts five GTA blocks to leverage the complementary information provided by different residual stages and achieves the best result.

**Comparison with Temporal Attention with Positional Embedding (TAPE)** Our GTA module is more effective in temporal modeling than TAPE because it not only considers the

$\text{res}_3$	$\text{res}_4$	$\text{res}_5$	Top-1
			17.0
+1			46.2
	+1		46.4
		+1	37.4
+1	+1		49.5
+2	+3		50.6

Table 8: Impact of inserting positions and number of blocks.

Model	w/o CCMH	w/ CCMH
+ TAPE	46.5	47.2
+ Pixel GTA	<b>48.0</b>	<b>48.5</b>
+ SA + TAPE	48.4	48.8
+ SA + Pixel GTA	<b>49.1</b>	<b>49.6</b>

Table 9: Ablation on positional embedding (TAPE) and cross-channel multi-head (CCMH) design.

Number of Regions	w/o RegionGTA	C	C/2	C/4	C/8	C/16	C/32
Top-1	49.6	49.7	50	50.3	<b>50.6</b>	50.3	50.1

Table 10: Impact of number of regions.  $C$  denotes the channel dimension of the feature map. Top-1 accuracy on SSv1 validation dataset are reported here.

Model	Top-1
Cross-channel Multi-head	<b>50.6</b>
Multi-head	50.1

Table 11: Comparison on cross-channel multi-head and multi-head.

chronological order of video frames but also models the temporal relationships among them. Results in Table 5 of the main paper show that GTA outperforms TAPE by **2.2%** on SSv1. Here, we provide a side-by-side comparison between TAPE and our Pixel GTA (without applying GTA to regions) in Table 9. Our Pixel GTA consistently outperforms TAPE under different settings. Furthermore, TAPE can also benefit from our cross-channel multi-head (CCMH) design, but Pixel GTA still achieves the best performance.

**Impact of number of regions.** We conduct experiments on the impact of the number of regions used in RegionGTA in Table 10. We can see that when increasing the number of regions from  $C/32$  to  $C$  ( $C$  is the channel dimension of the feature map), the accuracy increase first and reach the peak when  $K = C/8$ . More importantly, our RegionGTA consistently outperforms the model without RegionGTA under different values of  $K$ , which proves the effectiveness of our RegionGTA design.

**Comparison on cross-channel multi-head (CCMH) and multi-head.** In Table 11, we compare the performance of cross-channel multi-head and multi-head. We can see that the accuracy drops by 0.5% when the cross-channel design is removed from CCHM. It demonstrates that the channel interaction is also critical to help improve the accuracy of the action recognition task.

## C Relations to Prior Work

Our proposed decoupled framework and the cross-channel multi-head (CCMH) design are the two key differences between GTA and the prior work (GloRe [6]). Specifically, our Region GTA generates semantic regions within each frame and performs temporal modeling on each region *individually* along the time axis. In contrast, when applied to spatio-temporal

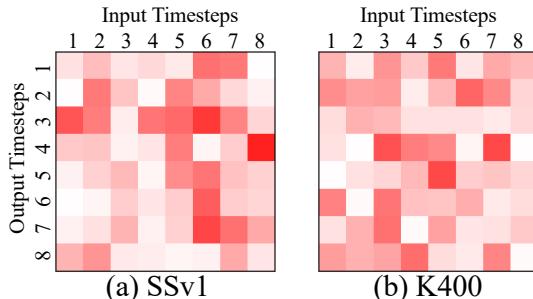


Figure 5: Visualization of global temporal attention weights on Something-Something v1 and Kinetics-400 datasets. Different columns represent timestamps of input from 1 to 8 and different rows represent timestamps of output from 1 to 8. Darker colors represent larger values of weights.

data, GloRe projects the whole 3D feature maps into semantic groups and models the interactions among them. We argue that this kind of grouping and modeling is not capable of capturing effective temporal relationships across different time steps. Moreover, GloRe leverages graph convolution to model node-wise interactions, which only considers information diffusion on each channel. Our GTA incorporates channel interactions to further improve temporal modeling, and we show its effectiveness in the experiments.

## D More Visualizations

**Visualization of Global Temporal Attention Weights** We provide visualization of the global temporal attention weights on two different datasets, Something-Something v1 and Kinetics-400 in Figure 5. Specifically, we average the learned global temporal attention weights across different groups and heads, and visualize the absolute value of attention weights. The darker colors represent larger values of weights. We can see that global attention weights of K400 and SSv1 are visually different. For SSv1, it tends to focus more on the latter part of the frames, while for Kinetics-400, the global temporal attention weights tend to focus more on the middle part of the frames. Our hypothesis is that because there are many action classes "pretending to do something", thus the latter part of the action are of vital importance to distinguish from "pretending to do something" vs "doing something". For example, for "pretending to pick something up" and "picking something up" actions, whether the object has been picked up eventually decides the action type. In addition, the global temporal attention weights are not flat across different timestamps, which verifies the effectiveness of our proposed GTA architecture.

**Visualization of Transformed Regions** We present visualization of the transformed regions in Figure 6. We can see that Region GTA can discover regions that share similar semantic meanings. For example, in the first video, the "ground" region and the "badminton" region are automatically identified, the "paper" and the "edge" are detected in the second video, and the "green gum" and the "hand" are obtained in the third video.

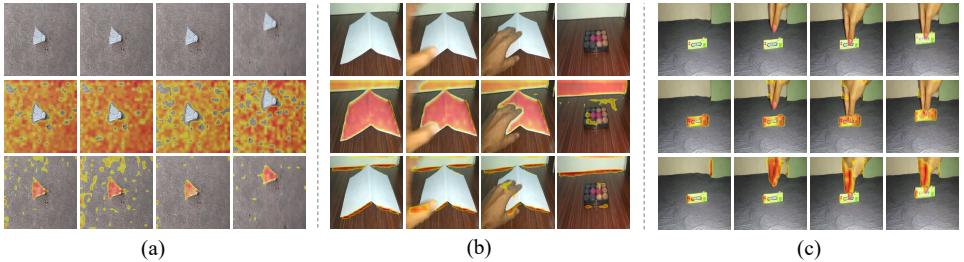


Figure 6: Visualization of the transformed regions of two examples: (a) “Turning the camera downwards while filming something”; (b) “Uncovering something”; (c) “Picking something up”. The first row is the frame sequences. The second and third rows are regions obtained by Region GTA.

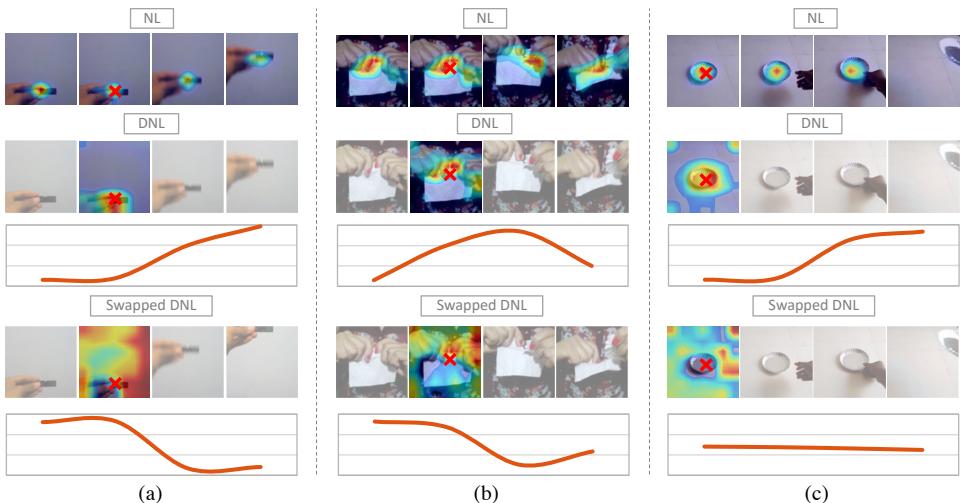


Figure 7: Visualization of the attention maps of three examples: (a) “Moving something up”; (b) “Tearing something into two pieces”; (c) “Picking something up”. The first row is the spatio-temporal attention map generated by the non-local module. The second and third row is the spatial and temporal attention map obtained by our decoupled non-local module. The fourth and fifth row is the spatial and temporal attention map generated by swapping the attention functions of the spatial and temporal attention block. The red cross mark denotes the query position.

**Visualization of Swapped Attention** To further verify that different context information is needed for spatial and temporal attention, we present the visualization of the swapped attention maps in Figure 7. Specifically, we swap the attention functions (i.e., query/key/value projections) of the spatial and temporal attention blocks and visualize the attention maps. We can see that after swapping the spatial and temporal attention functions, the generated temporal attention maps focus more on the frames with similar objects instead of the frames that are useful for recognizing the action.

For example, in Figure 7(a), the temporal attention weights are larger in the first two

frames which share a similar appearance with the same query position (i.e., the pen). Moreover, the spatial attention maps generated by the temporal attention functions also show substantially different patterns than the original ones. The visualization results further verify that different types of context information needed in spatial and temporal attention are captured in the decoupled non-local module.

## E Experiment Details

**Something-Something v1&v2 [16]** For the experiments based on the 2D CNN backbone, we follow the same sampling strategy as TSN [2] to sample 8 frames from the whole video. The same data augmentation is applied as TSN, which first resizes the input frames to  $240 \times 320$ , followed by the multi-scale cropping and random horizontal flipping. Note that we do not flip the clips which include the words “left” or “right” in their class labels (e.g., “pushing something from right to left”). We train the model for 50 epochs and start with a base learning rate of 0.01 with a batch size of 32. The first 2 epochs are used for linear warm-up [2] and the learning rate is reduced by a factor of 10 at 30, 40, 45 epochs. The backbone network is initialized with ImageNet pre-trained weights. For testing, we resize the input images to  $240 \times 320$  pixels and center crop  $224 \times 224$  pixels region. We sample 1 clip from each video for the experiments using 2D backbones.

For the experiments based on the 3D CNN backbone, we employ the same training and testing strategy as SlowFast- $16 \times 8$ -R50 [2]. We sample 16 and 64 frames for the slow and fast pathways, respectively.

**Kinetics-400 [8]** For the experiments using 2D CNN backbones, we adopt R2D-50 as the backbone and use 8 frames as input. The model is initialized with ImageNet pre-trained weights and trained with step-wise learning schedule following the PySlowFast codebase [2]. For the experiments using 3D CNN backbones, we use SlowFast- $8 \times 8$ -R101 that samples 8 and 32 frames for the slow and fast pathway, respectively. We first train the backbone model on Kinetics-400 and then fine-tune it with GTA, following the same practice for training the non-local blocks [2]. We sample 10 clips temporally and 3 crops spatially from each video for testing.