Group Contextualization for Video Recognition

Yanbin Hao
University of Science and Technology of China
Hefei, China
haoyanbin@hotmail.com

Hao Zhang*
Singapore Management University
Singapore
hzhang@smu.edu.sg

Chong-Wah Ngo
Singapore Management University
Singapore
cwngo@smu.edu.sg

Xiangnan He
University of Science and Technology of China
Hefei, China
xiangnanhe@gmail.com

Abstract

Learning discriminative representation from the complex spatio-temporal dynamic space is essential for video recognition. On top of those stylized spatio-temporal computational units, further refining the learnt feature with axial contexts is demonstrated to be promising in achieving this goal. However, previous works generally focus on utilizing a single kind of contexts to calibrate entire feature channels and could hardly apply to deal with diverse video activities. The problem can be tackled by using pair-wise spatio-temporal attentions to recompute feature response with cross-axis contexts at the expense of heavy computations. In this paper, we propose an efficient feature refinement method that decomposes the feature channels into several groups and separately refines them with different axial contexts in parallel. We refer this lightweight feature calibration as group contextualization (GC). Specifically, we design a family of efficient element-wise calibrators, i.e., ECal-G/S/T/L, where their axial contexts are information dynamics aggregated from other axes either globally or locally, to contextualize feature channel groups. The GC module can be densely plugged into each residual layer of the off-the-shelf video networks. With little computational overhead, consistent improvement is observed when plugging in GC on different networks. By utilizing calibrators to embed feature with four different kinds of contexts in parallel, the learnt representation is expected to be more resilient to diverse types of activities. On videos with rich temporal variations, empirically GC can boost the performance of 2D-CNN (e.g., TSN and TSM) to a level comparable to the state-of-the-art video networks. Code is available at https://github.com/haoyanbin918/Group-Contextualization.

*Hao Zhang is the corresponding author.

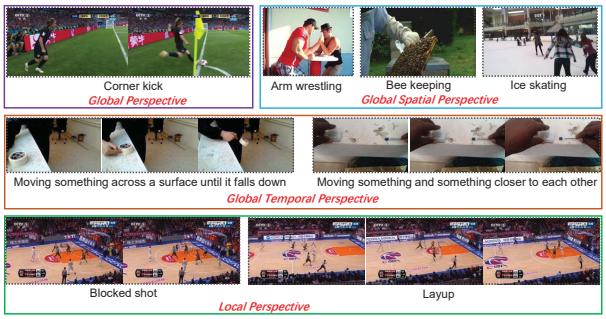

Figure 1. Perspective/axial preference of different video activities. The scene change caused by quick camera movement yearns for global context for recognizing the soccer highlight "*Corner kick*". "*Arm wrestling*", "*Bee keeping*" and "*Ice skating*" can be easily recognized even by a single keyframe. Whereas, the Something-Something activity examples (middle) rely much on temporal relations. The group activities, i.e., "*Blocked shot*" and "*Layup*", require a model to localize sub-activities.

1. Introduction

The 3D spatio-temporal nature of video signals allows video content to be flexibly analyzed from different perspectives or axes. Specifically, the signals can be transformed along various dimensions to capture the activities underlying a video. For example, in Figure 1, the soccer highlight "*Corner kick*" may require projection of the 3D video signal into a 1D vector to globally summarize the quick camera movement causing scene change. The less temporal activity categories "*Arm wrestling*", "*Bee keeping*" and "*Ice skating*" occurring in near static scenes prefer a 2D image representation to capture spatial perspective for classification. By contrast, the video events "*Moving something across a surface until it falls down*" and "*Moving something and something closer to each other*" require modeling of temporal relations over time axis. When it

is about classifying sub-activities such as "*Blocked shot*" and "*Layup*" in a lengthy basketball video, the localized 3D spatio-temporal analysis is preferred. These observations show the necessity of adjusting the features (C channels) of a 3-dimensional $T \times H \times W$ video tensor with a perspective aligned with video activities.

Feature contextualization [13, 16, 23, 29, 47, 50] is a technique that makes full use of axial contexts (e.g., spatial, temporal) to calibrate plain video features obtained from convolutional filters of CNN models (e.g., C3D [41], I3D [3], P3D [33]). Generally, axial contexts are referred to as information aggregated from other axes towards the features. For example, the global spatial context within the whole time length can be obtained by squeezing the tensor along time axis [50], while in contrast the global temporal context is acquired through shrinking along space axes [23, 29]. However, due to the large diversity of video activities, it is obvious that a single context cannot fit all activity cases. Given the activity "*Moving something and something closer to each other*" in Figure 1 as an example, globally aggregating contents along the time axis will harm the time order information, underemphasizing the subtle movement between objects. Also, the global aggregation will mess up sub-activities in the basketball highlights, diminishing the characteristics localized to sub-activities such as "dunk" and "foul". In these cases, the existing works [16,23,29,50], which focus on calibrating image/video features with only a specific global axial context, may under-perform due to the lack of versatility in representing various activities. On the other hand, projecting multiple axial contexts to a feature will increase the computation cost. Some works [13, 47] try to pairwisely attend each feature point of the 3D video tensor from local to global receptive field to adaptively decide the perspectives depending on context. Nevertheless, these works suffer from the heavy computation burden. The resulting network is not lightweight, and cannot densely plug into the existing network backbone. The most current work temporal difference network (TDN) [45] shows strong performance on activities that require both short-term and long-term dependencies through pairwisely computing the temporal differences with short and long intervals of time.

This paper addresses the limitation of feature contextualization for video recognition. Specifically, a novel feature contextualization paradigm, i.e, group contextualization (GC), is presented to derive feature representation that is generic to different activities and with lightweight computation. The GC module decomposes the channels into several paralleled groups and applies different feature contextualization operations on them respectively. As such, the calibrated feature is versatile for it integrating feature dynamics aggregated from different perspectives and potentially can recognize a wide variety of activities. The computational overload is kept to a minimum level by applying group convolution [30, 42, 49]. As each output channel only relates to the input channels within a group, only $\frac{1}{g}C^2$ (g is the number of divided groups) channel interactions is required, instead of C^2 of a standard convolution. Capitalizing on efficiency in computation, group convolution also allows us to analyze how a network exploits axial contexts in different layers for different activities.

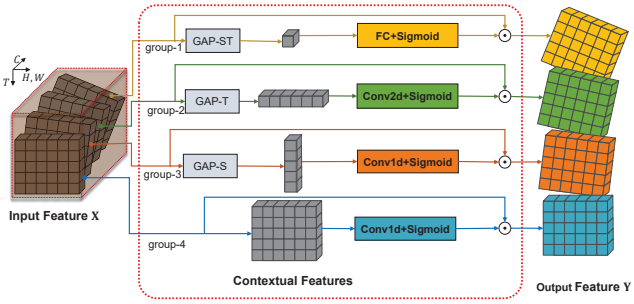

Figure 2. The workflow of the proposed group contextualization. "GAP" denotes global average pooling.

The workflow of GC module is illustrated in Figure 2. Particularly, the input CNN feature $\mathbf{X} \in \mathbb{R}^{T\times H\times W\times C}$ is split into four groups (group-1/2/3/4 with the size of $T \times H \times W \times \frac{1}{4}C$) along the channel dimension. To achieve separate contextualization, we accordingly design four element-wise calibrators[1] (ECals) to calibrate the four channel groups globally along space-time (ECal-G), globally along space (ECal-S), globally along time (ECal-T), and locally (ECal-L) in a small neighborhood in parallel. In this architecture, all ECals share the similar cascaded structure of "GAP/None+FC/Conv+Sigmoid" for efficiency, and achieve feature calibration with element-wise multiplication. Here, the global feature calibrations (ECal-G/S/T) perform feature pooling along a specific axis and contextualization on the different axis with the customized operators. In fact, the feature aggregation compresses the global information in an axis onto the other so that it enlarges the receptive field to the entire axial range. For example, ECal-T squeezes the input $T \times H \times W \times \frac{1}{4}C$ feature along the space axes, resulting in a $T \times 1 \times 1 \times \frac{1}{4}C$ contextual feature. In this case, when conducting temporal convolution on the resulted context feature, the global spatial content contributes the attention weight computation of each timestamp, which can benefit the recognition of video activities requiring long-range temporal relation (e.g., the video clips in Figure 1(middle)). In contrast, if we directly convolve the given feature map without any pooling operation, the global view narrows to a local neighbourhood (ECal-L). This perspective is particularly useful to localize sub-activities in lengthy video, e.g., "*Blocked shot*", "*Layup*" in the basketball video shown in Figure 1. Finally, by separately per-

[1]In this paper, feature adjustment, refinement and calibration, as well as their noun and verb forms, are used interchangeably.

forming feature refinement with ECal-G/S/T/L on those decomposed channel groups, we can reweight the input feature with multiple axial perspectives, resulting in a more discriminative representation $\mathbf{Y}$.

We summarize our contributions as below:

- **Group contextualization**. We propose a new regime named group contextualization (GC) for video feature refinement. GC encompasses a set of element-wise calibrators (ECal-G/S/T/L) to explicitly model multi-axial contexts and separately refine video feature groups in parallel.
- **Computation-efficient**. All ECal variants are designed in an efficient manner, and the channel decomposition as in group convolution moderates the extra computational cost incurred in feature calibration. For example, when averagely splitting the channels into four groups, GC only introduces 5.3%/1.3% extra parameters/FLOPs to the original TSN backbone.
- **Significant performance gain**. We verify that our GC module not only significantly improves the video recognition performance for several feedforward video networks (i.e., TSN, TSM and GST), but also can work together with the temporal difference network (TDN) leading to a notable performance gain.

2. Related Work

Since our work is mainly relevant to feature contextualization and group convolution techniques, we will separately review the related works from these two aspects.

Feature contextualization. Feature contextualization has been successfully demonstrated to be effective in image and video processing tasks, such as image retrieval/classification/segmentation [8, 11, 12, 16, 32, 43, 53, 57, 58], video/action recognition/classification [10, 13, 23, 29, 40, 47, 50]. Contextualization operation can enlarge the local receptive filed of a spatio-temporal filter to a global view with the support of perspective contexts. For example, the non-local neural network [47] recomputes the output of a local filter as a weighted sum of features of all points in the whole spatio-temporal video space. Since this operation needs pairwise comparison, its power is limited by the heavy computation burden. SSAN [10] factorizes the 3D pairwise attention into three separable spatial, temporal and channel attentions for efficiency. Another similar work is CBA-QSA [13] which omits the pairwise comparison and instead introduces a learnable query to guide the attention weight computing. To achieve more efficient feature contextualization, some approaches have studied modeling axial contexts through squeezing along specific axes. For example, SE-Net firstly proposes the squeeze-and-excitation mechanism to work as a self-gating operator to elementwisely refine image features with global context. The gather-excite network (GE-Net) [15] generalizes SE-Net by investigating various levels of spatial context granularity. S3D-G [50] brings the feature refinement idea of SE-Net [16] to calibrate the features of S3D with the global axial context. TEA [23] introduces a motion excitation module to calculate pixel-wise movement of subsequent frames and a multiple temporal aggregation module to enlarge the temporal receptive field with the aggregated temporal axial context. TANet [29] also performs average pooling to collapse spatial information towards time axis but additionally considers long-range temporal modeling by having a feed-forward neural network as a separate branch to refine the features. Compared to prior works, GC not only takes the global/temporal axial context into account for long-range temporal modeling but also considers the underlying video activities for feature contextualization.

Group convolution. The group convolution [14, 21, 30, 42, 49, 55] divides the feature maps into small groups and uses multiple kernels to separately compute their channel outputs. This leads to not only much lower computation loads but also wider networks helping to learn a varied set of low level and high level features. In video processing area, the work [14], which directly replaces the spatial 2D convolutional kernels of 2D-CNNs such as ResNext [49] and DenseNet [18] with 3D counterparts, explores the potential of 3D group convolutions for video recognition. The channel-separated convolutional network (CSN) [42] studies various settings of 3D group convolution as well as its extreme version depthwise convolution on C3D [41] for efficient video classification. More recently, the grouped spatial-temporal network (GST) [30] proposes to decompose the feature channels into two asymmetric groups and uses 2D and 3D convolutions to separately learn the spatial and temporal information. The gate-shift module (GSM) [37] extends temporal shift module used in [26, 52] with learnable shift parameters and uses the channel decomposition to further reduce parameters. The above group convolution works focus on designing generic models, while our proposed group contextualization is to recalibrate the plain video feature with multiple axial contexts for enhancing the off-the-shelf neural network models for video recognition.

3. Group Contextualization

The group contextualization module is constructed as a plug-and-play module, which can be used to calibrate any given 4D $T \times H \times W \times C$ video tensor. In this section, we first elaborate the details of GC module, as well as four designed element-wise calibrators, in a general manner. Then, we integrate it into four representative video CNN models for enhancing their capacity of representation learning and give analysis to model complexity. Finally we examine the impact of varying channel positions in the backbones.

GC aims to calibrate a portion of channels of a video fea-

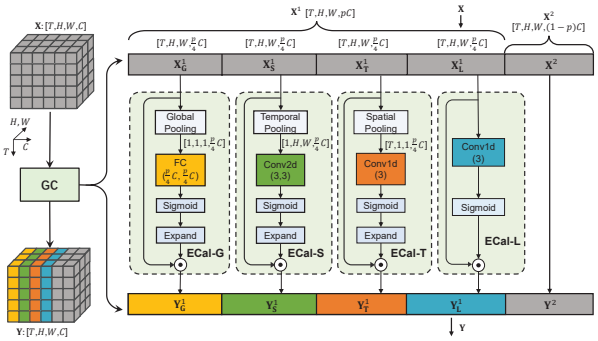

Figure 3. The schema of the GC module. "⊙" denotes the Hadamard product.

ture using a specific axial context at a time. The schema of GC module is illustrated in Figure 3. Suppose that a 4D feature tensor is $\mathbf{X} \in \mathbb{R}^{T\times H\times W\times C}$ yielded by a convolutional operator or counterpart, where T, H, W, C denote time-length, space-height, space-width and channel-size, respectively. Firstly, GC splits the feature tensor into two groups with a partition ratio p along the channel dimension, resulting in $\mathbf{X}^1 \in \mathbb{R}^{T\times H\times W\times pC}$ and $\mathbf{X}^2 \in \mathbb{R}^{T\times H\times W\times(1-p)C}$. Then, four feature calibrators are customized to focus on four different axial perspectives and separately refine the four feature channel subgroups of $\mathbf{X}^1$, i.e., $\mathbf{X}^1_{G/S/T/L} \in \mathbb{R}^{T\times H\times W\times \frac{p}{4}C}$ in parallel, resulting in four corresponding outputs $\mathbf{Y}^1_{G/S/T/L} \in \mathbb{R}^{T\times H\times W\times \frac{p}{4}C}$. Finally, the calibrated feature parts and the non-calibrated feature part are concatenated along channel dimension, and the output of GC is

$$\mathbf{Y} = \text{Concat}(\mathbf{Y}^1_G, \mathbf{Y}^1_S, \mathbf{Y}^1_T, \mathbf{Y}^1_L, \mathbf{Y}^2), \tag{1}$$

where $\mathbf{Y} \in \mathbb{R}^{T\times H\times W\times C}$. Next, we present the designs of four element-wise calibrators in detail.

3.1. Element-wise Calibrators

Global-wise Calibrator (ECal-G). The ECal-G block instantiates the global axial context through globally pooling the 4D $\mathbf{X}^1_G$ across time and space, yielding a contextual vector with the size of $\frac{p}{4}C$. Then, to make use of the aggregated contextual information, a fully-connected (FC) layer is to compute the interactions among channels of the vector. Finally, a Sigmoid function is employed to calculate the channel-wise gating weights and an Expand operation further inflates the weight vector to the same size of $\mathbf{X}^1_G$ by element copying. Formally, the calculation flow can be as follows

$$\mathbf{Y}^1_G = \text{Expand}(\text{Sigmoid}(\text{FC}(\frac{1}{T\times H\times W}\sum_{t,h,w}\mathbf{X}^1_G[t,h,w])))\odot\mathbf{X}^1_G. \tag{2}$$

Spatial-wise Calibrator (ECal-S). ECal-S block shrinks the input tensor along the temporal axis using an average pooling operation, resulting in a $1\times H\times W\times C$ contextual feature. A 2D convolution with 3×3 kernel is then adopted to compute the impact to a local spatial neighbor. Similarly, the Sigmoid and Expand operations are for the element-wise weighting tensor generation. Finally, we have

$$\mathbf{Y}^1_S = \text{Expand}(\text{Sigmoid}(\text{Conv2d}(\frac{1}{T}\sum_t \mathbf{X}^1_S[t,:,:])))\odot\mathbf{X}^1_S. \tag{3}$$

Temporal-wise Calibrator (ECal-T). In contrast to ECal-S, ECal-T pools the input along the spatial axes, aggregating the global spatial content into $T\times 1\times 1\times C$ statistics. Feature contextualization is achieved by using a temporal 1D convolution, which can mix the global spatial information within a local temporal receptive field. Further passing to the Sigmoid and Expand functions, the refined output tensor $\mathbf{Y}^1_G$ is computed as

$$\mathbf{Y}^1_T = \text{Expand}(\text{Sigmoid}(\text{Conv1d}(\frac{1}{H\times W}\sum_{h,w}\mathbf{X}^1_T[:,h,w])))\odot\mathbf{X}^1_T. \tag{4}$$

Local-wise Calibrator (ECal-L). Since ECal-L focuses on capturing local contexts within a neighboring field, we directly utilize a convolutional unit to achieve the local interaction computation. In the implementation, a 1D temporal convolution with $3\times 1\times 1$ kernel is adopted. This is because that a simple 1D convolution requires much lighter computational load than a 3D convolution, and temporal modeling is more critical in video feature learning. Without the use of global average pooling operation, the size of input tensor in ECal-L is kept during the weight calculation. Hereby, we have

$$\mathbf{Y}^1_L = \text{Sigmoid}(\text{Conv1d}(\mathbf{X}^1_L))\odot\mathbf{X}^1_L. \tag{5}$$

The above four ECals work individually and follow the self-gating regime for feature calibration. They compute the element-wise gating weights by contextualizing a specific axial perspective of interest. The element-wise gating weights could be global-wise (ECal-G),spatial-wise (ECal-S), temporal-wise (ECal-T), and local-wise (ECal-L). As a result, the proposed GC module can achieve multiple perspectives of feature contextualization in parallel for a single input.

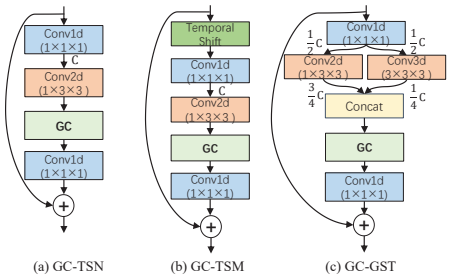

Figure 4. Integration of GC module into TSN, TSM and GST networks.

3.2. Network Architecture and Model Complexity

We integrate the GC module into three basic video networks, i.e., TSN [46] (a standard 2D spatial model),

TSM [26] (a 2D temporal shift model), and GST [30] (a 3D group convolution model), and a more advanced network, i.e., TDN [45] (modeling both short-term and long-term dependencies by temporal differences), referred to as GC-TSN, GC-TSM, GC-GST and GC-TDN. Figure 4 shows the integrated blocks in GC-TSN/TSM/GST. The GC module is inserted after the 2D/3D convolution layer. Particularly, since TDN uses different stages to separately model the short-term (the first two stages) and long-term temporal information (the latter three stages), we thus do not visualize its integrated block. The implementation of GC-TDN follows GC-TSN.

Block	Params	Percentage	
		$p = \frac{1}{2}$	$p = 1$
Residual block (TSN)	$17 \times C^2$	100.0%	
ECal-G	$\frac{1}{16} \times p^2 \times C^2$	0.09%	0.37%
ECal-S	$\frac{9}{16} \times p^2 \times C^2$	0.83%	3.31%
ECal-T	$\frac{3}{16} \times p^2 \times C^2$	0.28%	1.10%
ECal-L	$\frac{3}{16} \times p^2 \times C^2$	0.28%	1.10%
Total	$p^2 \times C^2$	1.47%	5.88%

Table 1. Comparison of parameters for different calibrator blocks. For simplicity, bias terms and batchnorm terms are omitted in parameter counting.

The partition ratio p controls the portion of channels to be calibrated and hence governs the complexity of GC module. Table 1 lists the number of parameters of each ECal as well as their sums. To make a clearer comparison, we also show the number of parameters of the original residual block of TSN. It is worth noting that TSN and TSM have the same model complexity as the temporal shift operation in TSM is computationally free. Specifically, the parameters introduced by GC module is as low as 1.47% of the original 2D Residual block when $p = \frac{1}{2}$.

3.3. Does Channel Position Make Any Difference?

Except the partition ratio p of channels, the position of channel groups could be another discussible variable in GC. To be specific, we empirically investigate a new position setting, i.e., the loop version, as shown in Figure 5. Unlike the standard version that remains the channel group positions unchanged in all residual blocks of ResNet, the loop version periodically shifts the channel groups being calibrated block by block with a step of 1. Consequently, the channel groups adjusted by calibrators keep varying in position among residual blocks. In the experiment, we observe different performance tendencies with different backbones. Specifically, we find that there is no significant performance difference between the standard version and the loop version on TSN, TSM and TDN, but the loop version gains a great performance improvement (from 45.6% to 46.7% with 8 frames on Something-Something V1 dataset) on GST. This may be because that the convolutional features in TSN/TSM/TDN are entangled together across channels and each subgroup can thus contain similar information

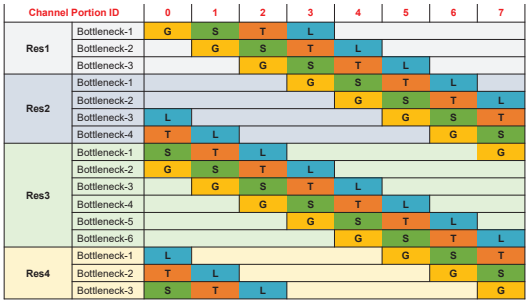

Figure 5. Loop version of GC. The standard 2D ResNet-50 is adopted as the backbone. The partition ratio p is set to $\frac{1}{2}$. G, S, T, L denotes ECal-G, ECal-S, ECal-T and ECal-L units respectively.

(e.g., spatial information in TSN, spatio-temporal information in TSM and short/long-term temporal difference in TDN). Whereas, in GST the spatial and temporal features are separately learnt without any channel interaction between groups. Since the channel groups in GST provide different types of features, keeping on utilizing the same calibrator to adjust fixed channels may be not optimal. For example, when setting $p = \frac{1}{2}$, the spatio-temporal feature group ($\frac{1}{4}C$) outputted by the Conv3d in GST will not participate in the feature calibration in the standard version. If we change p to 1, the spatial feature group ($\frac{3}{4}C$) is fixed to perform the calibration with ECal-G/S/T and the other spatio-temporal feature group is only involved in ECal-L. Differently, the loop version skillfully allows both the two feature groups of GST to achieve feature calibration with all the four ECals. Based on this, we apply the standard version on TSN/TSM/TDN and the loop version on GST in implementation. It is worth mentioning that both the standard and loop versions have the same model complexity.

4. Experiment

We conduct experiments on different benchmarks, including Something-Something V1&V2 [9, 31] and Kinetics-400 [20] for video recognition. The four video datasets cover a broad range of activities. Specifically, the **Something-Something V1&V2** datasets show 174 fine-grained humans performing pre-defined activities and are more focused on modeling the temporal relationships. The **Kinetics-400** dataset covers 400 human action classes with less motion variations. Due to space limitation, we include the results on EGTEA Gaze+ [25], which is a dataset offering first-person videos, Diving48 [24] with unambiguous dive sequence, and Basketball-8&Soccer-10 [13] with group activities in the supplementary document.

4.1. Experimental Setup

We insert the GC module into four different 2D and 3D ResNets, including TSN, TSM, GST and TDN. Most experiments are based on the backbone of ResNet-50 pretrained on ImageNet [34]. Notably, we additionally add a "Batch-

Norm" layer after each convolutional/FC layer in the ECals for TSN, TSM and TDN. We implement GC-Nets in Pytorch and run them on servers with 4×2080Ti or 4/8×3090.

Training & Inference. The **training** protocol mainly follows the work [46]. Specifically, we use uniform sampling for all datasets. The spatial short side of input frames is resized to 256 maintaining the aspect ratio and then cropped to 224×224. Data augmentation also follows [46]. Training configurations for GC-TSN/TSM/GST are set as follows: a batch-size of 8/10 per GPU, an initial learning rate of 0.01 for 50 epochs and decayed at epoch 20 and 40, the SGD optimizer. GC-TDN follows the training protocol of TDN [45]. The dropout ratio is set to 0.5. During the **inference**, we uniformly sample 8 frames per video and use the 224×224 center crop for performance report in the ablation study. In the final performance comparison, we sample multiple clips per video and take no more than three crops per clip. Specifically, the test protocols are: 2 clips × 3 crops (224×224) for Something-Something V1&V2, and 1 clip × 1 center crop (224×224) for others. We will also specify the sampled frames in the tables.

4.2. Ablation Study

We present ablation study to investigate the effect of hyperparameters, including the channel partition ratio p, channel position, calibrator variants and backbones, on Something-Something V1 dataset.

p and calibrators. We first compare different ECals on TSN with $p = \frac{1}{2}, 1$. The four types of ECals are designed to calibrate video feature with different axial context concerns. And the channel partition ratio p is introduced to control the number of channels to be calibrated by ECals. As shown in Table 2, we observe that ECal variants, regardless of their types, consistently improve the recognition performance of the backbone TSN, indicating their effectiveness. Although varying the value of p from $\frac{1}{2}$ to 1 will result in slight increase of model size and computational cost, the performance boost is noticeable (e.g., 26.3%→27.3% for ECal-G and 35.9%→36.4% for ECal-T).

Channel position and backbones. Secondly, we test both the standard and loop GC versions on the four backbones. Here, we also set $p = \frac{1}{2}, 1$. Table 2 shows their results. Compared to the single calibrator, the GC module, which combines the four ECals in parallel, achieves much better performance on TSN. The GC-TSM, GC-GST and GC-TDN also gain significant performance improvement (45.6%→48.9% for TSM, 44.4%→46.7% for GST, 52.3%→53.7% for TDN) to their original backbones. Consistently, the models with larger $p = 1$ outperform their counterparts with $p = \frac{1}{2}$. Based on the above results, we fix $p = 1$ for the GC-Nets in this work. For the channel position, we observe different performance tendencies on the four backbones, i.e., the result of loop version is clearly

Backbone	Calibrator	(p, Channel)	Params	FLOPs	Top-1 (%)
TSN	—	—	23.9M	32.9G	19.7
	SE3D	—	26.4M	32.9G	27.8 (+8.1)
	GE3D-G	—	23.9M	32.9G	22.3 (+2.6)
	GE3D-C	—	25.2M	33.3G	44.2 (+24.5)
	S3D-G	—	25.1M	32.9G	28.0 (+8.3)
	NLN	—	31.2M	49.4G	30.3 (+10.6)
	ECal-G	$(\frac{1}{2}, \frac{1}{8}C)$	23.9M	32.9G	26.3 (+6.6)
		$(1, \frac{1}{4}C)$	23.9M	32.9G	27.3 (+7.6)
	ECal-T	$(\frac{1}{2}, \frac{1}{8}C)$	23.9M	32.9G	35.9 (+16.2)
		$(1, \frac{1}{4}C)$	24.1M	32.9G	36.4 (+16.7)
	ECal-S	$(\frac{1}{2}, \frac{1}{8}C)$	24.0M	32.9G	34.0 (+14.3)
		$(1, \frac{1}{4}C)$	24.6M	33.0G	34.1 (+14.4)
	ECal-L	$(\frac{1}{2}, \frac{1}{8}C)$	23.9M	33.0G	44.8 (+25.1)
		$(1, \frac{1}{4}C)$	24.1M	33.2G	44.9 (+25.2)
	GC	$(\frac{1}{2}, \frac{1}{2}C)$	24.2M	33.0G	47.1 (+27.4)
		$(1, C)$	25.1M	33.3G	47.9 (+28.2)
		$(1, C)$, loop	25.1M	33.3G	**48.0 (+28.3)**
TSM	—	—	23.9M	32.9G	45.6
	SE3D	—	26.4M	32.9G	46.7 (+1.1)
	GE3D-G	—	23.9M	32.9G	45.7 (+0.1)
	GE3D-C	—	25.2M	33.3G	47.0 (+1.4)
	S3D-G	—	25.1M	32.9G	46.8 (+1.2)
	NLN	—	31.2M	49.4G	47.2 (+1.6)
	GC	$(\frac{1}{2}, \frac{1}{2}C)$	24.2M	33.0G	48.7 (+3.1)
		$(1, C)$	25.1M	33.3G	**48.9 (+3.3)**
		$(1, C)$, loop	25.1M	33.3G	**48.9 (+3.3)**
GST	—	—	21.0M	29.2G	44.4
	GC	$(\frac{1}{2}, \frac{1}{2}C)$	21.4M	29.3G	45.5 (+1.1)
		$(1, C)$	22.3M	29.6G	45.6 (+1.2)
		$(1, C)$, loop	22.3M	29.6G	**46.7 (+2.3)**
TDN	—	—	26.1M	36.0G	52.3
	GC	$(1, C)$	27.4M	36.7G	**53.7 (+1.4)**
		$(1, C)$, loop	27.4M	36.7G	53.6 (+1.3)

Table 2. Performance changes using different backbones, calibrators, partition ratio p and channel position on Something-Something V1 dataset. "Channel" denotes the number of channels in each calibrated feature group. SE3D is the 3D variant of SE-Net [16] by replacing the 2D spatial average pooling with the 3D spatio-temporal average pooling. GE3D-G and GE3D-C are two 3D variants of GE-Net [15], where GE3D-G adopts global average pooling and GE3D-C employs 3D depthwise convolution. Their architectures can be found in Appendix. NLN denotes the nonlocal [47] module.

better than the standard version on GST, and their performances are about the same on TSN, TSM and TDN. As analysed in Section 3.3, this is because that feature channels in TSN, TSM and TDN are entangled together during the feature learning while the group convolution method GST separately models the spatial and temporal features.

Comparison with other calibrators. Thirdly, we integrate the 3D variants of SE-Net [16] and GE-Net [15], i.e., SE3D and GE3D-G/C, S3D-G and NLN, into the TSN and TSM backbones. Their hyperparameters are set as the same to their original papers. The NLN-Nets follows the implementation of [26]. From Table 2, we can find that our GC module far outstrips SE3D, GE3D-G and S3D-G which only consider the global context and the pairwise self-attention NLN when using TSN as backbone. Since GE3D-C uses three 3D depthwise convolution layers to model local spatio-temporal context, relatively good performance (44.2% Top-1 accuracy) is attained on TSN but still lower than our GC (47.9%). On TSM, our GC can outper-

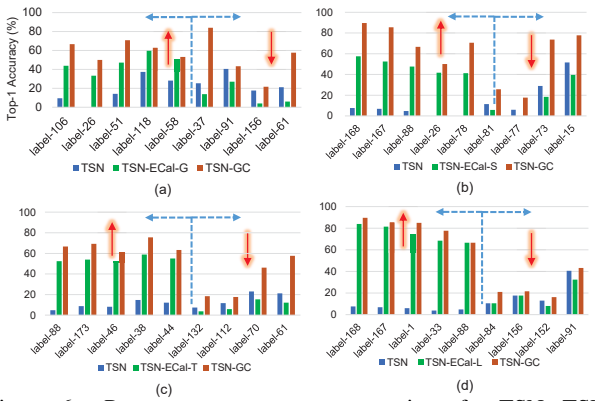

Figure 6. Per-category accuracy comparison for TSN, TSN-ECals and GC-TSN over several selected activity categories on Something-Something V1 dataset (validation set). In each subfigure, we show 5 action categories that are improved most by TSN-ECal and another 4 activity categories that are being degraded by TSN-ECal but improved by GC-TSN. See the supplementary material for the label details of these selected activities.

form the five other calibrators by the margins of 1.7%-3.2%. Moreover, compared to the self-attention NLN module that results in 31% extra parameters and 50% extra FLOPs to the backbones, our GC only introduces as low as 5% extra parameters and 1.2% extra FLOPs.

4.3. Example Demonstration

We show the per-category results of TSN-ECal variants and GC-TSN to understand the impact of axial contexts on different kinds of video activities in Figure 6. Specifically, ECal-G can boost the recognition of activities that need global contexts, e.g., "*label-106: Putting something in front of something*" in Figure 6(a). ECal-S focuses on enhancing the feature spatial-wisely with the context aggregated along temporal dimension and thus can improve the performance for activities that require more spatial than temporal information, for example "*label-26: Lifting a surface with something on it but not enough for it to slide down*" in 6(b). TSN-ECal-T and TSN-ECal-L, which naturally calibrate the feature globally and locally along temporal dimension respectively, can significantly improve the recognition performance for activities that require temporal reasoning, as shown in the first 5 categories in figures 6(c) and (d). However, those failure cases in the four subfigures (last 4 terms) provide evidence that one kind of axial contexts is not suitable for all activity categories. For example, TSN-ECal-G fails to model the activities that involve rich spatial-temporal interactions between objects, e.g., "*label-61: Pouring something into something until it overflows*", and TSN-ECal-S under-performs on "*label-73: Pretending to put something into something*" and "*label-81: Pretending to squeeze something*" which require strong temporal reasoning. Encouragingly, the GC module that aggregates all the four ECals indeed alleviates these shortcoming of individual ECal, leading to performance improvement for a variety of activities.

Method	Params	#Frame	FLOPs×Clips	Top-1	Top-5
CorrNet [44]	—	32	115.0G×10	49.3	—
TIN [36]	24.6M×2	8+16	101G×1	49.6	78.3
V4D [54]	—	8×4	167.6G×30	50.4	—
SmallBig [22]	—	8+16	157G×1	50.4	80.5
TANet [29]	25.1M×2	8+16	99.0G×1	50.6	79.3
STM [19]	24.0M	16	33.3G×30	50.7	80.4
TEA [23]	—	16	70.0G×30	52.3	81.9
TEINet [27]	30.4M×2	8+16	99.0G×1	52.5	—
RNL-TSM [17]	35.5M×2	8+16	123.5G×2	52.7	—
GST* [30]	21.0M	16	58.4G×2	46.2	75.0
GST* [30]	21.0M×2	8+16	87.6G×2	48.6	78.3
TSM [26]	23.9M	16	65.8G×2	48.4	78.1
TSM [26]	23.9M×2	8+16	98.7G×2	50.3	79.3
TDN [45]	26.1M	16	72.0G×1	53.9	82.1
TDN [45]	26.1M×2	8+16	108.0G×1	55.1	82.9
GC-GST	22.3M	8	29.6G×2	48.8	78.5
GC-GST	22.3M	16	59.1G×2	50.4	79.4
GC-GST	22.3M×2	8+16	88.7G×2	52.5	81.3
GC-TSN	25.1M	8	33.3G×2	49.7	78.2
GC-TSN	25.1M	16	66.5G×2	51.3	80.0
GC-TSN	25.1M×2	8+16	99.8G×2	53.7	81.8
GC-TSM	25.1M	8	33.3G×2	51.1	79.4
GC-TSM	25.1M	16	66.5G×2	53.1	81.2
GC-TSM	25.1M×2	8+16	99.8G×2	55.0	82.6
GC-TSM	25.1M×2	8+16	99.8G×6	55.3	82.7
GC-TDN	27.4M	8	36.7G×1	53.7	82.2
GC-TDN	27.4M	16	73.4G×1	55.0	82.3
GC-TDN	27.4M×2	8+16	110.1G×1	**56.4**	**84.0**

Table 3. Comparison of performance on Something-Something V1 dataset. "*" indicates that the result is obtained by ourselves.

4.4. Comparison with the State-of-the-Arts

We compare GC-Nets with state-of-the-art networks in this section. The result comparison follows the same protocol of using RGB frames as input and adopting ResNet50 unless otherwise specified.

Something-Something V1 & V2. A comprehensive comparison between our GC-Nets and SOTAs on Something-Something V1&V2 datasets are presented. Tables 3 and 4 list the comparison in terms of Top-1/5 accuracy, FLOPs and model complexity. GC-TDN achieves the highest Top-1 accuracies of 56.4% and 67.8% with (8+16) frames × 1 clip on Something-Something V1&V2, respectively, which outperform all the CNN-based SOTAs by large margins (1.3%-36.7% for V1 and 0.8%-37.8% for V2). Moreover, all GC-Nets, including GC-GST, GC-TSN, GC-TSM and GC-TDN, consistently outperform their backbone networks with significant performance gains, demonstrating the capacity of GC module in recognizing diverse activities and the strong versatility against various deep video networks. For example, GC-TSN boosts the original TSN model with an absolute improvements of 28.2% (19.7%→47.9%) on V1 and 32.4% (30.0%→62.4%) on V2 with the same 8-frame input. Equipping TSN, which is an image-based CNN, with GC empowers the modeling of temporal relationship between objects on V1&V2. The GC module can also improve the more advanced TDN by 1.3% on V1 and 0.8% on V2 with the same 8+16 frames. This demonstrates that the axial contexts modeled by GC

Method	Params	#Frame	FLOPs×Clips	Top-1	Top-5
TIN [36]	24.6M	16	67.0G×1	60.1	86.4
RubiksNet [5]	8.5M	8	15.8G×2	61.7	87.3
TSM+TPN [51]	—	8	33.0G×1	62.0	—
SlowFast [7]	32.9M	4+32	65.7G×6	61.9	87.0
SlowFast(R101) [7]	53.3M	8+32	106G×6	63.1	87.6
SmallBig [22]	—	16	114.0G×6	63.8	88.9
STM [19]	24.0M	16	33.3G×30	64.2	89.8
TEA [23]	—	16	70.0G×30	65.1	89.9
TEINet [27]	30.4M×2	8+16	99.0G×1	65.5	89.8
TANet [29]	25.1M×2	8+16	99.0G×6	66.0	90.1
TimeSformer-HR [2]	121.4M	16	1703G×3	62.5	—
ViViT-L [1]	352.1M	32	903G×4	65.4	89.8
MViT-B [4]	36.6M	64	455G×3	67.7	90.9
Video-Swin-B [28]	88.8M	16	321G×3	**69.6**	**92.7**
TSN [56] from [26]	23.9M	8	32.9G×1	30.0	60.5
GST* [30]	21.0M	8	29.2G×2	59.8	86.3
GST* [30]	21.0M	16	58.4G×2	61.7	87.2
GST* [30]	21.0M×2	8+16	87.6G×2	63.1	88.3
TSM [26]	23.9M	8	32.9G×2	61.2	87.1
TSM [26]	23.9M	16	65.8G×2	63.1	88.2
TSM [26]	23.9M×2	8+16	98.7G×2	64.3	89.0
TDN [45]	26.1M	8	36.0G×1	64.0	88.8
TDN [45]	26.1M	16	72.0G×1	65.3	89.5
TDN [45]	26.1M×2	8+16	108G×1	67.0	90.3
GC-GST	22.3M	8	29.6G×2	61.9	87.8
GC-GST	22.3M	16	59.1G×2	63.3	88.5
GC-GST	22.3M×2	8+16	88.7G×2	65.0	89.5
GC-TSN	25.1M	8	33.3G×2	62.4	87.9
GC-TSN	25.1M	16	66.5G×2	64.8	89.4
GC-TSN	25.1M	8+16	99.8G×2	66.3	90.3
GC-TSM	25.1M	8	33.3G×2	63.0	88.4
GC-TSM	25.1M	16	66.5G×2	64.9	89.7
GC-TSM	25.1M×2	8+16	99.8G×2	66.7	90.6
GC-TSM	25.1M×2	8+16	99.8G×6	67.5	90.9
GC-TDN	27.4M	8	36.7G×1	64.9	89.7
GC-TDN	27.4M	16	73.4G×1	65.9	90.0
GC-TDN	27.4M×2	8+16	110.1G×1	**67.8**	**91.2**

Table 4. Comparison of performance on Something-Something V2 dataset. "*" indicates that the result is obtained by ourselves.

can work cooperatively with the temporal difference contexts used by TDN. Compared to the more sophisticated Transformer-based models like MViT and Video-Swin, the GC-TSM and GC-TDN obtain lower Top-1 accuracies. The performance is compensated by lower computational cost and model complexity. GC-TDN requires 110.1G FLOPs, which is about 11.4 times cheaper than MViT-B (1,365G FLOPs) and 7.7 times lower than Video-Swin-B (963G FLOPs).

Kinetcis-400. We report the results of GC-TSN with 8 frames and GC-TSM/TD with 8 and 16 frames respectively, in Table 5. Firstly, the GC-TSN/TSM/TDN improve the Top1 accuracy upon TSN/TSM/TDN by 4.6%/2.0%/1.2% under the same input, respectively. They are much significant in terms of the data scale of Kinetics-400 dataset. Secondly, GC-TDN, with 16×30 clips as input, achieves the 79.6% Top1 accuracy, which is the highest one among the competing methods. This result is much better than other models equipped with feature contextualization techniques, such as Nonlocal-I3D, S3D-G and TEA, which further demonstrates the superior performance of the proposed GC module.

Model	Params	#Frame	FLOPs×Clips	Top1	Top5
I3D (InceptionV1) [3]	—	64	—	72.1	90.3
Nonlocal-I3D [47]	35.3M	32	282G×10	74.9	91.6
S3D-G (InceptionV1) [50]	—	64	71.4G×30	74.7	**93.4**
TEA [23]	—	16	70G×30	76.1	92.5
TEINet [27]	30.8M	16	66G×30	76.2	92.5
TANet [29]	25.6M	16	86G×12	76.9	92.9
SmallBig [22]	—	8	57G×30	76.3	92.5
SlowFast(8×8) [7]	32.9M	8+32	65.7G×30	77.0	92.6
X3D-L [6]	6.1M	16	24.8G×30	77.5	92.9
TSN [56]	24.3M	8	32.9G×10clip	70.6	89.2
TSM [26]	24.3M	16	66.0G×10	74.7	91.4
TDN [45]	26.6M	8+16	108.0G×30	78.4	93.6
GC-TSN	25.6M	8	33.3G×10	75.2	92.1
GC-TSM	25.6M	8	33.3G×10	75.4	91.9
GC-TSM	25.6M	16	66.6G×10	76.7	92.9
GC-TSM	25.6M	16	66.6G×30	77.1	92.9
GC-TDN	27.4M	8	36.7G×30	77.3	93.2
GC-TDN	27.4M	16	73.4G×30	78.8	93.8
GC-TDN	27.4M	8+16	110.1G×30	**79.6**	94.1

Table 5. Comparison of performance on Kinetics-400 dataset.

5. Conclusion

We have presented the regime of group contextualization, which aims at deriving robust representations generic to various video activities by calibrating plain features computed from off-the-shelf networks with multiple contexts. The family of element-wise calibrators is designed to work on different grouped feature channels independently. The group operation results in a much lower computation cost increase (5.3%/1.3% extra parameters/FLOPs) and substantial performance improvements (0.4%-32.4%) to backbones. More surprisingly, when GC module is integrated into the 2D spatial TSN model, GC-TSN achieves absolute 28.2%/32.4% performance improvements on Something-Something V1/V2 and even performs much better than the advanced 3D spatio-temporal GST and TSM models. We conclude that since the videos in Something-Something datasets contain rich global/local human-object interactions, GC module that explores various global/local spatial/temporal axial contexts to calibrate the original feature exhibits excellent performance. Similar results are also observed from the other datasets (e.g., Diving and Kitchen Activities). Moreover, compared to the other feature calibration methods, such as SE3D, GE3D, S3D-G, TEA and TANet that only use a single context, GC-Nets consistently achieve better performances, which further proves the feasibility and advantages of the proposed group contextualization. The significant performance improvement of GC-TDN further demonstrates that our GC can also work together with the other temporal difference context (TDN).

Acknowledgements

The work was supported in part by the National Natural Science Foundation of China (No. 62101524), by the National Key Research and Development Program of China under Grants 2020YFB1406703, and by the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 grant.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. *arXiv preprint arXiv:2103.15691*, 2021. 8

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021. 8, 14

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 2, 8, 14

[4] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *arXiv preprint arXiv:2104.11227*, 2021. 8

[5] Linxi Fan, Shyamal Buch, Guanzhi Wang, Ryan Cao, Yuke Zhu, Juan Carlos Niebles, and Li Fei-Fei. Rubiksnet: Learnable 3d-shift for efficient video action recognition. In *European Conference on Computer Vision*, pages 505–521. Springer, 2020. 8

[6] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 203–213, 2020. 8

[7] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6202–6211, 2019. 8

[8] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019. 3

[9] 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 *Proceedings of the IEEE International Conference on Computer Vision*, pages 5842–5850, 2017. 5

[10] Xudong Guo, Xun Guo, and Yan Lu. Ssan: Separable self-attention network for video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12618–12627, 2021. 3

[11] Yutian Guo, Jingjing Chen, Hao Zhang, and Yu-Gang Jiang. Visual relations augmented cross-modal retrieval. In *Proceedings of the 2020 International Conference on Multimedia Retrieval*, pages 9–15, 2020. 3

[12] Ning Han, Jingjing Chen, Guangyi Xiao, Hao Zhang, Yawen Zeng, and Hao Chen. Fine-grained cross-modal alignment network for text-video retrieval. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3826–3834, 2021. 3

[13] Yanbin Hao, Hao Zhang, Chong-Wah Ngo, Qiang Liu, and Xiaojun Hu. Compact bilinear augmented query structured attention for sport highlights classification. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 628–636, 2020. 2, 3, 5, 12, 13, 14

[14] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018. 3

[15] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: exploiting feature context in convolutional neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 9423–9433, 2018. 3, 6, 12

[16] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 2, 3, 6, 12

[17] Guoxi Huang and Adrian G Bors. Region-based non-local operation for video classification. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10010–10017. IEEE, 2021. 7

[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 3

[19] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2000–2009, 2019. 7, 8

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

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 3

[22] Xianhang Li, Yali Wang, Zhipeng Zhou, and Yu Qiao. Smallbignet: Integrating core and contextual views for video classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1092–1101, 2020. 7, 8

[23] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 909–918, 2020. 2, 3, 7, 8

[24] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 513–528, 2018. 5, 12

[25] Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 619–635, 2018. 5, 12, 14

[26] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings*

of the IEEE/CVF International Conference on Computer Vision, pages 7083–7093, 2019. 3, 5, 6, 7, 8, 14

[27] Zhaoyang Liu, Donghao Luo, Yabiao Wang, Limin Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Tong Lu. Teinet: Towards an efficient architecture for video recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11669–11676, 2020. 7, 8

[28] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021. 8

[29] Zhaoyang Liu, Limin Wang, Wayne Wu, Chen Qian, and Tong Lu. Tam: Temporal adaptive module for video recognition. *arXiv preprint arXiv:2005.06803*, 2020. 2, 3, 7, 8

[30] Chenxu Luo and Alan L Yuille. Grouped spatial-temporal aggregation for efficient action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5512–5521, 2019. 2, 3, 5, 7, 8, 14

[31] Farzaneh Mahdisoltani, Guillaume Berger, Waseem Gharbieh, David Fleet, and Roland Memisevic. On the effectiveness of task granularity for transfer learning. *arXiv preprint arXiv:1804.09235*, 2018. 5

[32] Diganta Misra, Trikay Nalamada, Ajay Uppili Arasanipalai, and Qibin Hou. Rotate to attend: Convolutional triplet attention module. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3139–3148, 2021. 3

[33] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *proceedings of the IEEE International Conference on Computer Vision*, pages 5533–5541, 2017. 2

[34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 5

[35] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 12

[36] Hao Shao, Shengju Qian, and Yu Liu. Temporal interlacing network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11966–11973, 2020. 7, 8

[37] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Gate-shift networks for video action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1102–1111, 2020. 3

[38] Swathikiran Sudhakaran and Oswald Lanz. Attention is all we need: Nailing down object-centric attention for egocentric activity recognition. *arXiv preprint arXiv:1807.11794*, 2018. 14

[39] Hao Tan, Jie Lei, Thomas Wolf, and Mohit Bansal. Vimpac: Video pre-training via masked token prediction and contrastive learning. *arXiv preprint arXiv:2106.11250*, 2021. 14

[40] Yi Tan, Yanbin Hao, Xiangnan He, Yinwei Wei, and Xun Yang. Selective dependency aggregation for action classification. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 592–601, 2021. 3

[41] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 2, 3

[42] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5552–5561, 2019. 2, 3

[43] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2017. 3

[44] Heng Wang, Du Tran, Lorenzo Torresani, and Matt Feiszli. Video modeling with correlation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 352–361, 2020. 7

[45] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. Tdn: Temporal difference networks for efficient action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1895–1904, 2021. 2, 5, 6, 7, 8, 14

[46] 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 *European conference on computer vision*, pages 20–36. Springer, 2016. 4, 6

[47] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 2, 3, 6, 8, 14

[48] Xiaohan Wang, Yu Wu, Linchao Zhu, and Yi Yang. Symbiotic attention with privileged information for egocentric action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12249–12256, 2020. 14

[49] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 2, 3

[50] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 305–321, 2018. 2, 3, 8, 12

[51] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 591–600, 2020. 8

[52] Hao Zhang, Yanbin Hao, and Chong-Wah Ngo. Token shift transformer for video classification. In *Proceedings of the*

29th ACM International Conference on Multimedia, pages 917–925, 2021. 3

[53] Hao Zhang and Chong-Wah Ngo. A fine granularity object-level representation for event detection and recounting. *IEEE Transactions on Multimedia*, 21(6):1450–1463, 2018. 3

[54] Shiwen Zhang, Sheng Guo, Weilin Huang, Matthew R Scott, and Limin Wang. V4d: 4d convolutional neural networks for video-level representation learning. *arXiv preprint arXiv:2002.07442*, 2020. 7

[55] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018. 3

[56] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 803–818, 2018. 8, 14

[57] Bin Zhu and Chong-Wah Ngo. Cookgan: Causality based text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5519–5527, 2020. 3

[58] Bin Zhu, Chong-Wah Ngo, Jingjing Chen, and Yanbin Hao. R2gan: Cross-modal recipe retrieval with generative adversarial network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11477–11486, 2019. 3