Hierarchical Hourglass Convolutional Network for Efficient Video Classification

Yi Tan
University of Science and Technology of China, Hefei, China
ty133@mail.ustc.edu.cn

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

Hao Zhang
Singapore Management University
Singapore
zhanghaoinf@gmail.com

Shuo Wang
University of Science and Technology of China, Hefei, China
shuowang.hfut@gmail.com

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

ABSTRACT

Videos naturally contain dynamic variation over the temporal axis, which will result in the same visual clues (e.g., semantics, objects) changing their scale, position, and perspective patterns between adjacent frames. A primary trend in video CNN is adopting spatial-2D convolution for spatial semantics and temporal-1D convolution for temporal dynamics. Though the direction achieves a favorable balance between efficiency and efficacy, it suffers from misalignment of visual clues with large displacements. Particularly, rigid temporal convolution would fail to capture correct motions when a specific target moves out of the reception field of temporal convolution between adjacent frames.

To tackle large visual displacements between temporal neighbors, we propose a new temporal convolution named *Hourglass Convolution* (HgC). The temporal reception field of HgC has an hourglass shape, where the spatial reception field is enlarged in prior & post temporal frames, enabling an ability to capture large displacement. Moreover, since videos contain long, short-term movements viewed from multiple temporal interval levels, we hierarchically organize the HgC net to both capture temporal dynamics from frame (short-term) and clip (long-term) levels. Besides, we also adopt strategies, such as low-resolution for short-term modeling and channel reduction for long-term modeling, from efficiency concerns. With HgC, our H^{2}CN equips off-the-shelf CNNs with a strong ability in capturing spatio-temporal dynamics at a neglectable computation overhead. We validate the efficiency and efficacy of HgC on standard action recognition benchmarks, including Something-Something V1&V2, Diving48, and EGTEA Gaze+. We also analyse the complementarity of frame-level motion and clip-level motion with visualizations. The code and models will be available at https://github.com/ty-97/H2CN.

✉Corresponding author. Xiangnan He is also affliated with the Institute of Dataspace, Hefei Comprehensive National Science Center.

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3547841

CCS CONCEPTS

• **Computing methodologies** → **Activity recognition and understanding**.

KEYWORDS

Video classification, convolution, neural network, attention

ACM Reference Format:
Yi Tan, Yanbin Hao✉, Hao Zhang, Shuo Wang, and Xiangnan He✉. 2022. Hierarchical Hourglass Convolutional Network for Efficient Video Classification. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3503161.3547841

1 INTRODUCTION

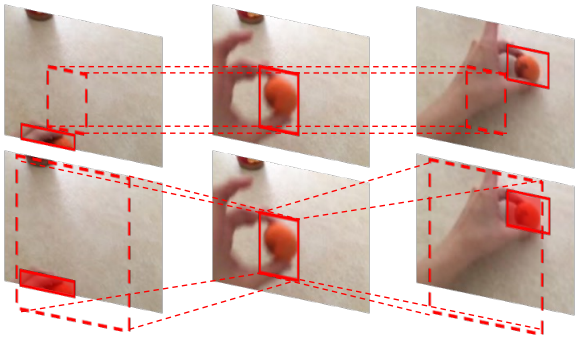


Figure 1: Sampled clips from Something-Something V1. The recognition of "*put a table tennis ball on flat*" needs to focus on the table tennis ball and hand, as well as their interaction. Conventional temporal convolution fails to capture the correct motion due to misalignment of visual clues caused by the large displacement of the ball and hand.

Visual clues (e.g., semantics, objects) evolve with the temporal axis, changing their scale, position, and perspective patterns. These dynamic variations aggregate into discriminative motion patterns and are crucial for video content classification.

Towards capturing these evidential dynamical patterns, existing works could be divided into three directions:

- *Optical Flow*, which explicitly extracts motions out of RGB frames is used intuitively as external information to enhance dynamics

modeling in video actions. Representative works include the two-stream network [40], which represents motion in optical-flow [20] form, and independently feeds static (RGB) and dynamic (opt-flow) into two separate CNNs. Predictions of two streams are further late-fused. Though effective, the two-stream network suffers from a heavy computation burden due to the per-fetching of opt-flow modality and adding an extra CNN branch.
- *Conventional Temporal Convolutions*, which perform temporal aggregation at the same spatial location of temporal neighbors. Specifically, conventional 1D temporal convolutions are combined with 2D spatial convolution in a cascade [3, 45] or parallel manner [37, 47, 54]. Equipping 2D-CNN with the ability of temporal perception, these paradigms gain favorability in network design. However, their temporal modeling capability is limited without special consideration of the temporal dimension.
- *Attention Strategies*, which introduce similarities between spatio-temporal variations to represent motion pattern [25, 51]. Though effective, it shares the same computation issue as opt-flow-based methods since pairwise similarities calculation between spatial-temporal localities is inefficient.

This paper argues that visual displacements between adjacent temporal frames prevent the rigid 1D temporal convolution from capturing motion patterns well. For example, in Figure 1, the action "*pick up a table tennis ball and put the ball on the table*" includes interactions of core objects like "*hand*" and "*tennis ball*". As the time flows, the spatial semantics of individual frames gradually change from "*pick up the ball*" to "*lift the ball in the air*" and "*put it on the table*". In this progress, the scale, position, and patterns of "*hand*" and "*tennis ball*" change. Rigid 1D temporal convolution does not take a large variation into consideration as it only aligns the same spatial location across different timestamps for dynamical modeling. Thereby, it could easily lose core visual clues when evidential objects move out of the reception field in adjacent frames.

To tackle large visual displacements between temporal neighbors, we propose a new temporal convolution named Hourglass Convolution (HgC). With the hourglass-shaped reception field, i.e., larger spatial reception fields for prior and latter temporal neighbors, our HgC enables more accurate motion capturing, especially for those with large movements. Besides, motions in video activities involve multiple granularities simultaneously. They behave with various magnitudes on different temporal interval levels, e.g., frame level and clip level.

With the concerns above, we extend the HgC to capture both frame-level and clip-level motions hierarchically. We name our video CNN as Hierarchical Hourglass Convolutional Network (H^2CN). Specifically, we operate HgC on frames within a clip and aggregate features across adjacent frames for frame-level motion modeling. Frame-level motion information then supply the shallow 2D convolution features with the intra-clip temporal dynamics via a lateral connection. For efficiency consideration, we conduct frame-level motion modeling on a lower frame resolution. As for clip-level motion modeling, we employ HgC across clips to aggregate the long-term temporal dynamics. Consequently, a local clip will be contextualized by long-term temporal perception. To balance computations, we performed HgC on features preprocessed with dimensional reduction. From frame-level and clip-level motion modeling, H^2CN captures diverse motion patterns and equips off-the-shelf CNNs with a strong ability in modeling spatial-temporal dynamics at a neglectable computation overhead.

Our HgC presents a new temporal convolution that can better capture the motion pattern with its strong capability to model large visual displacements caused by visual clues movement. Thanks to the Hierarchical structure, the proposed H^2CN can capture both frame-level and clip-level motions. To show the effectiveness of HgC and H^2CN, we conduct extensive experiments on real-world video classification benchmarks, such as Something-Something V1 & V2, Diving48, and EGTEA Gaze+. The experimental results show that the convolution scheme of HgC outperforms the rigid 1D temporal convolution. Also, H^2CN shows a strong ability of motion representation learning and obtains SOTA performances compared with other competing methods. Our contributions are briefly summarized as below.

- **Hourglass Convolution**. We propose a new hourglass convolution operator which sets larger spatial reception fields for prior & post time-stamps than the current one to better model the spatio-temporal dynamics in the video.
- **Hierarchical Hourglass Convolutional Network.** We build a new network for action recognition with the hourglass convolution operator. It can capture motion features at frame and clip levels and demonstrate solid discriminative power.
- **SOTA Efficiency and Efficacy**. Our proposed H^2CN demonstrates superior performance on four video benchmarks, covering a broad range of video activities, to other SOTA methods but incurs little computation overhead (i.e., 1%/3% extra parameters/FLOPs) to the ResNet backbone.

2 RELATED WORKS

Convolution structure. With the current availability of powerful parallel machines (e.g., GPUs, TPUs) and large amounts of training data, convolution has witnessed great progress in a large range of deep learning communities, such as computer vision [19, 57] and natural language processing [13]. Convolutional operators usually have rigid reception fields to fit the shape of the studied modality (e.g., square image and language sequence). In the image processing area, there have been some convolution variants like deformable convolutions [5, 6] that augments the 2D spatial sampling locations with additional offsets to allow the network to obtain information away from its regular local neighborhood. In video processing area, researchers try to build high-dimensional convolutions (e.g., decomposed 3D [37, 47, 54], 3D [3, 46] and even 4D [58] convolutions) to facilitate the spatio-temporal tensor calculating. Although these video convolutions demonstrate better results than 2D convolution on video motion modeling, their abilities are limited to model large visual displacements between temporal neighbors due to the same spatial reception filed for each temporal stamp.

Motion capturing. Except for the above implicit motion capturing with high-dimensional operations, efforts to explicitly model the motion pattern are spared. Particularly, TSM [30] proposes a temporal shift of partial channels to enhance a pure 2D CNN with motion capturing ability. GSM [41] extends TSM with learnable shift parameters and uses the channel decomposition to compress

parameters. Moreover, RubikShift [9] even attempts to replace all convolutional filters with lightweight spatial/temporal shift operations. Token shift transformer further [55] explores the transfer of shift operator in transformer structures. Although efficient, these methods may neglect the long-range temporal characteristics of video modality, which deserves special consideration. As a more intuitive representation of motion information, optical flow [20] is proved to be effective in video recognition [40]. However, the pre-computation for optical flow is time-consuming. Researchers turn to adopt temporal difference [27, 31, 49] as an approximation of optical flow. Like two sides of a coin, optical flow and temporal difference are unaware of the static appearance feature, which also is crucial in video recognition.

Spurred by the promising performance, researchers learn motion information with various attention-based methods [15, 16, 18, 54]. For example, S3D-G [54] aggregates the global Spatio-temporal context and assigns it to feature channels by using the squeeze-and-excitation operation as SE-Net [21]. Apart from the long-range axial contexts, SDA [44] and GC [17] further consider the local spatio-temporal contexts for accurate motion modeling. Some works utilize self-attention [4, 10, 48] for CNN feature refinement. For instance, Non-local [51] represents a neural response at a specific location as the weighted sum of features from all Spatio-temporal voxels based on the feature similarity. STSS [25] generates motion by learning self-similarity in a local Spatio-temporal window. Patrick *et al.* [36] model the similarity across frames as motion trajectory in transformer architecture. Similarity-based models perform well with the inherent non-locality for motion modeling, but the external pairwise similarity calculation makes them less efficient.

3 METHOD

In this section, we will elaborate on our proposed H^{2}CN. The core of H^{2}CN is the newly designed hourglass convolution (HgC) which provides a new paradigm for temporal information aggregation. Based on HgC, we propose to model complex motions of videos on both frame and clip levels. Firstly, we briefly revisit the rigid temporal convolution. Secondly, we describe our HgC in detail, which sets different sizes of reception fields for different time offset positions. Finally, we present a hierarchical video recognition network H^{2}CN using HgC as key motion capturing operations.

3.1 Revisiting Temporal Convolution

We first revisit the temporal convolutions, which is widely used in 3D spatio-temporal convolution [3, 45] (implicitly) and decomposed 3D convolution [37, 47, 54]. For clearly understanding the working flow, we adopt a $3 \times 1 \times 1$ depth-wise temporal convolution[1] as the representative of temporal convolutions. Given a video feature input $\mathbf{X}$ with the size of $T \times H \times W \times C$ where $\{T, H, W\}$ denote the dimensions of {time, space-x, space-y} and C is the channel number ($C = 1$ for calculation convenience), the feature response of the (t, h, w)-th voxel can be represented as:

$$\tilde{x}_{t,h,w} = \alpha_{-1} \cdot x_{t-1,h,w} + \alpha_0 \cdot x_{t,h,w} + \alpha_1 \cdot x_{t+1,h,w}, \tag{1}$$

[1]The depth-wise convolution does not perform channel interaction and thus is easy to understand. Here, the spatial kernel size is set 1 only for computational simplicity. Other kernel sizes like 3, 5, 7 can also be used.

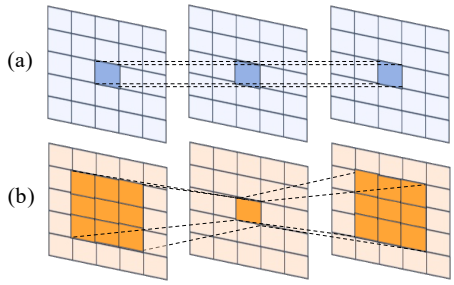


Figure 2: Comparison between (a) traditional temporal convolution and (b) hourglass convolution.

where $\boldsymbol{\alpha} = [\alpha_{-1}, \alpha_0, \alpha_1]$ is the parameter of the temporal convolution. We can find that the temporal convolution essentially acts as a learnable pixel-level feature aggregator for the same spatial location of temporal adjacent frames/clips. However, the spatio-temporal interactions not always happen in the same location. As the video plays, spatio-temporal interactions change their location, scale and pattern. Hence, the simply feature aggregator at the same location may fail to model the complete motion information which is of most importance in video recognition.

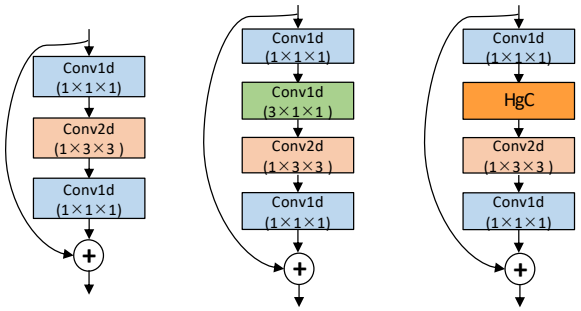
3.2 Hourglass Convolution

Considering the issue of 1D temporal convolution mentioned in section 3.1, we believe that it would benefit the motion modeling if setting a larger reception field (i.e., kernel) for frames/clips far away from the anchor (the middle one). That is, the spatial kernel sizes of different timestamps should not be kept the same. In this work, we propose a new temporal convolution termed as hourglass convolution (HgC), which is named for its hourglass shaped reception field. Specifically, the kernel size on a timestamp is determined by its relative time offset. Suppose that the size of temporal reception field is K (i.e., a total of K frames or clips) and the time offset is $i \in [-\lfloor \frac{K}{2} \rfloor, \lfloor \frac{K}{2} \rfloor]$. The size of spatial reception field in HgC is thus computed as $(p \cdot |i|+1, p \cdot |i|+1)$, where p denotes the slope of reception field expansion. For example, when setting $K = 3$ and $p = 2$, the spatial kernel sizes become $\{(3, 3), (1, 1), (3, 3)\}$ for $\{t-1, t, t+1\}$-th frames respectively. In the implementation, we adopt a two-step operation. Firstly, we use a 2D convolution or average-pooling with the determined kernel size to achieve information aggregation in space. Secondly, we use a 1D convolution with kernel 3 to perform temporal information aggregation along the time axis. Formally, given the input $\mathbf{X}$, the output of HgC can be written as:

$$HgC(\mathbf{X})_{t,h,w} = \sum_{i=-\frac{K}{2}}^{\frac{K}{2}} \alpha_i \cdot f(\mathbf{X}_{t+i,:,:}; W_{p\cdot|i|+1,p\cdot|i|+1})_{h,w}, \tag{2}$$

where $f(\cdot)$ denotes the spatial aggregation operation (2D convolution or 2D average-pooling), and $W_{p|i|+1,p|i|+1}$ represents the size of expanded reception field.

Figure 2 illustrates the comparison between traditional temporal convolution and HgC, where the temporal reception field is set as 3 for both of them and p is set as 2. It can be found that our HgC has larger spatial reception filed for temporal neighbors (e.g., $(3, 3)$ vs. $(1, 1)$ of the counterpart), which significantly reduces the risk of



(a) TSN (b) R(2+1)D (c) HgC-ResNet

Method	$f(\bullet)$	Top-1	#P	FLOPs
TSN	——	19.7	23.9M	32.9G
R(2+1)D	——	46.0	23.9M	32.9G
HgC-ResNet	AvgPool2D	46.4	23.9M	32.9G
	Conv2D	**47.0**	23.9M	33.1G

Figure 3: Network structure of (a) TSN (2D ResNet), (b) R(2+1)D (decomposed 3D ResNet) and (c) HgC-ResNet as well as their performance comparison on the validation set of Something-Something V1. Two reception field expansion strategies are investigated in HgC.

object vanishing from the reception filed. Consequently, HgC will be more suitable for motion modeling.

As a tapas, empirically we compare HgC with a typical motion model operation R(2+1)D [47] in video recognition task. R(2+1)D cascades a 2D spatial convolution and 1D temporal convolution for spatio-temporal information modeling. The spatial reception fields of R(2+1)D remain unchanged among all time offsets in the convolutions. For a fair comparison, similar to R(2+1)D network, we also use the 2D ResNet-50 as backbone and densely plugin HgC before the 2D convolution, referred to as HgC-ResNet. Here, we instantiate f with both a 2D depth-wise convolution and the simple average-pooling operation. The experiment is conducted on the commonly used Something-Something V1 dataset. Figure 3 shows their performance comparison w.r.t top-1 recognition accuracy, model parameters (#P) and FLOPs as well as their network structures. As observed, both R(2+1)D and HgC-ResNet can significantly improve the performance of 2D CNN backbone (TSN). HgC-ResNet variants, regardless of the types of spatial operation, consistently outperform R(2+1)D by remarkable margins (e.g., 0.4% with 2D average-pooling and 1.0% with 2D convolution) but with almost the same computation cost. This comparison primarily shows a proof of the good capability of HgC in video motion modeling.

3.3 Hierarchical Hourglass Convolutional Network

As analysed in section 1, magnitudes of video motions vary in temporal intervals. In this part, we explore two levels of motion tempos, i.e., frame-level and clip-level movements, and model them by separately applying HgC on consecutive frames and video clips. The overall architecture is shown in Figure 4. Specifically, given a video $\mathbf{V}$, we firstly divide it into T non-overlapped&equal clips $\{V_1, V_2, \cdots, V_T\}$. Then, a frame is randomly sampled from each clip, resulting in a total of T keyframes $\mathbf{F} = [F_1, F_2, \cdots, F_T]$ with each one having the size of $H \times W \times 3$. The clip-level temporal dynamics are thus reserved in these keyframes. We refer them to as clip-level motions. In practice, a single static keyframe cannot completely express the entire clip content since there often exist rich micro dynamics within a clip. Therefore, we additionally sample another 4 frames centered at each keyframe F_t. Finally, a total of 5 frames $\mathbf{C}_t = [C_t^{-2}, C_t^{-1}, F_t^0, C_t^1, C_t^2]$ are sampled to represent the t-th clip. Below, we first elaborate on the frame-level motion capturing from a clip (i.e., $\mathbf{C}_t$) and then detail the clip-level motion capturing from the video (i.e., $\mathbf{F}$).

Frame-level motion capturing. We conduct frame-level motion capturing on each video clip represented by $\mathbf{C}_t \in \mathbb{R}^{5\times H\times W\times 3}$, for which a frame-level motion capturing block (FMCB) is designed. Our goal is to learn the micro motions from the 5 consecutive frames and preserve these motions information into a frame-size feature tensor $\mathbf{M}_t^{fm} \in \mathbb{R}^{1\times H\times W\times C}$. As shown in Figure 5(a), FMCB consists of two HgCs which are connected in series. Here, normalization and activation functions are omitted for notation convenience. The computation flow of FMCB can be defined as follows

$$\mathbf{H}_t^{fm} = HgC_2(HgC_1(DownSample(\mathbf{C}_t))), \quad (3)$$

$$\mathbf{M}_t^{fm} = UpSample(Conv2d(\mathbf{H}_t^{fm}; 7\times 7)). \quad (4)$$

The $DownSample$ function is used to resize the frame resolution (H, W) to a lower size $(\frac{H}{2}, \frac{W}{2})$ and the $UpSample$ recoveries the original resolution. The two HgCs, i.e., HgC_1 and HgC_2, achieve micro motion modeling. Without feature padding, the temporal length of $\mathbf{H}_t^{fm}$ is reduced to 1 for fitting the fusion with the appearance feature of keyframe F_t. Afterwards, a 2D convolution with kernel 7×7 is used to further readjust the learned micro motions within relatively larger spatial neighbors, resulting in the ultimate feature tensor $\mathbf{M}_t^{fm}$. Finally, we fuse $\mathbf{M}_t^{fm}$ and the appearance feature of the keyframe F_t by elementwise addition. After the above processing, the frame-level dynamics modeled by FMCB are incorporated entirely in the clip.

Clip-level motion capturing. FMCB enhances the appearance feature of each keyframe (clip) F_t with frame-level motions within a clip in the early stage. However, it is incapable of exploring the long-range motion pattern of the entire video because of the limitation in the temporal reception field. Consequently, the clip-level motion capturing block (CMCB) is proposed to model long-range motion structure across the T clips. Here, CMCB is designed as plug-in attention module. Specifically, we denote the video feature outputted by a specific convolution layer in the backbone as $\mathbf{Y} \in T\times H\times W\times C$. For computational efficiency, we firstly introduce a convolution layer with kernel $1 \times 1 \times 1$ followed by a normalization layer, to reduce the dimensions of channel C controlled by a hyperparameter r_c (following previous work [21], we set r_c as 16). After getting the light weighted clip-level feature $\mathbf{Y}' \in T \times H \times W \times \frac{C}{r_c}$, we utilize HgC upon $\mathbf{Y}'$ to aggregate the motion dependency across clips:

$$\mathbf{D}^{cm} = HgC(\mathbf{Y}'). \quad (5)$$

Then, we use a global average pooling layer to squeeze the spatial information and focus on the temporal dimension:

$$\mathbf{D}^{cmp} = AvgPool2d(\mathbf{D}^{cm}). \quad (6)$$

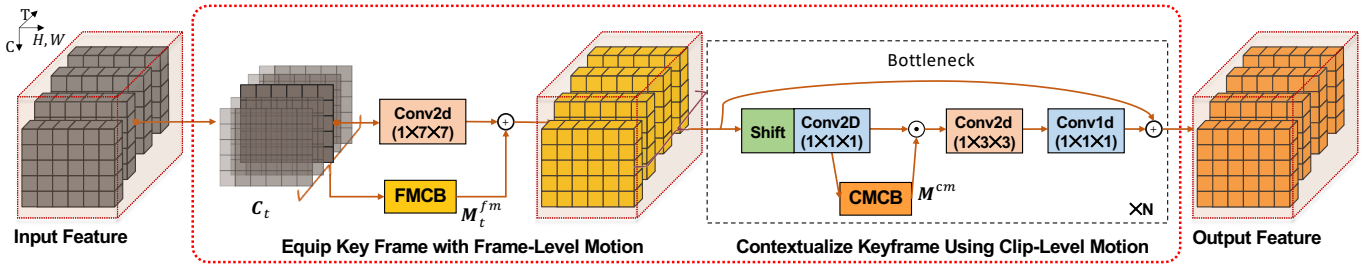


Figure 4: The overall architecture of Hierarchical Hourglass Convolutional Network.

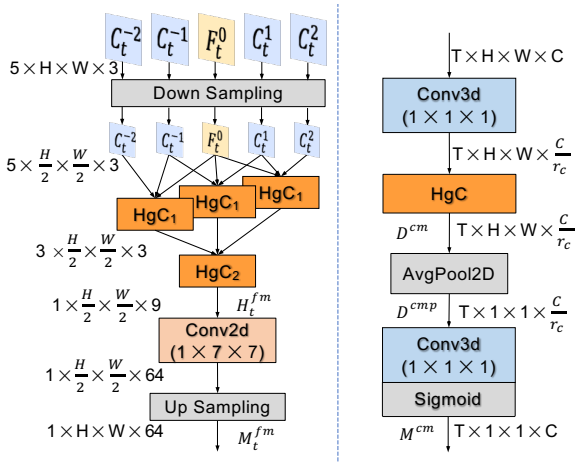


Figure 5: The work flows of frame-level motion capturing block (left) and clip-level motion capturing block (right).

To excite the dependency context for clips, we recover the channel dimension of $\mathbf{D}^{cmp}$ to the original value C using a convolution layer with kernel $1 \times 1 \times 1$. Finally, the dependency feature is activated by a Sigmoid function and used to elementwisely refine the original feature $\mathbf{Y}$ in a gating manner:

$$\mathbf{M}^{cm} = Sigmoid(Conv3d(\mathbf{D}^{cmp}; 1 \times 1 \times 1)), \tag{7}$$

$$\mathbf{Z} = \mathbf{M}^{cm} \odot \mathbf{Y}. \tag{8}$$

Since FMCB and CMCB are designed to capture motion information on different temporal interval levels, the magnitudes of motions between consecutive frames and between clips are disparate. We empirically ablate multiple kernel settings of HgC in the experiments.

3.4 Integrated Model

Our proposed FMCB and CMCB can be easily integrated into off-the-shelf CNN backbones. Here, we use ResNet as an example. The FMCB is laterally connected to the $Conv1$ layer to achieve frame-level motion modeling and fusion. The CMCB is densely inserted into each bottleneck block (between the first and second convolutional layer) of $Res1-4$ stages. As such, we have equipped the backbone with both short-range (frame-level) and long-range (clip-level) dependencies by the proposed FMCB and CMCB, resulting in the new video network H^{2}CN. Benefiting from the use of low-resolution and channel reduction, H^{2}CN introduces as low as 1% parameters and 3% FLOPs of the backbone.

4 EXPERIMENTS

To verify the effectiveness of HgC and hierarchical framework we conduct extensive ablation studies in 4.3. We further compare our H^{2}CN with SOAT methods including CNN-based approaches as well as more sophisticated transformer-based methods to justify the superiority of H^{2}CN in 4.4.

4.1 Datasets

We conduct extensive experiments on four benchmark datasets, including Something-Something V1&V2 [14, 35], Kinetics-400 [60], Diving48 [28] and EGTEA Gaze+ [29] for video recognition. The metrics are top-1 and top-5 precision. **Something-Something V1&V2** datasets contain 174 fine-grained humans performing predefined actions with objects and focus on more temporal dynamics than spatial statics. **Kinetcs-400** is a large-scale video dataset which covers 400 categories in daily life. Kinetics dataset mainly focuses on static appearance. In addition, we include two datasets with relative smaller scale, i.e., EGTEA Gaze+ which provides first-person videos and Diving 48 with dive sequence. Covering a broad range of actions in videos, these five datasets can not only evaluate the effectiveness but also the robustness of our proposed model. Due to the space limitation, we include the performance comparison on Diving48 and EGTEA Gaze+ in the supplementary material.

4.2 Implementation Details

We use ResNet as the backbone to implement our H^{2}CN framework. All models are implemented with Pytorch toolkit and run on 8 × 3090 GPUs.

Training. Following the common setting [50], we uniformly sample 8 or 16 keyframes from input videos for all datasets. As for resolution, we resize the sampled frames into 240 × 320 images and then crop a 224 × 224 patch out of the resized frames for Something-Something V1&V2. For Kinetics-400, Diving48 and EGTEA Gaze+, we resize the short-side of frames to 256 maintaining the aspect ratio and then crop a 224×224 patch out of resized frames. Data augmentations such as random scaling before cropping and random horizontal flipping are also adopted.

We train the network with batch size 8 per GPU and we set the learning rate (lr) as 0.01. The total training epoch is set as 100 for Kinetics-400 and 60 for other datasets. We decay lr by 0.1 at 50 75 90 for Kinetics-400 and decay lr at epoch 30, 45 and 55 for other datasets. The dropout ratio is set as 0.5. The backbone ResNet is pre-trained on ImageNet.

Inference. We sample 8 or 16 keyframes per video, resize them into 240 × 320 images for Something-Something dataset and 256 ×

Table 1: Performance comparison of different reception field expansion slopes p for HgC in CMCB on the validation set of Something-Something V1.

method		top-1	top-5	#p	FLOPs
w/o FMCB		45.6	74.2	23.9M	32.9G
FMCB	p=2	52.3	80.3	23.9M	33.6G
	p=4	**52.5**	80.5	23.9M	33.6G
	p=6	52.3	80.3	23.9M	33.6G

256 for Diving48 and EGTEA Gaze+, then we use one center crop with size 224 × 224 from the resized images. Testing augmentations are specified in tables.

4.3 Ablation Study

In this section, we investigate the effectiveness of frame-level motion capturing block (FMCB) and clip-level motion capturing block (CMCB) as well as the p value which stands for the slope of reception field expansion for prior and latter temporal neighbors. In addition, we ablate our HgC and rigid 1D temporal convolution in FMCB and CMCB. These evaluations are conducted on the validation set of Something-Something V1 dataset. We use a center crop of 8 keyframes for model testing in all ablation studies and the metrics are Top1/Top5 acc., model size and FLOPs are also specified.

Firstly, we investigate the effectiveness of FMCB and CMCB with different p values. The magnitude of p determines the size of reception field for prior and latter temporal neighbors. We empirically set p as 2, 4, 6, the reception field for prior and latter temporal neighbors are then respectively calculated as 3×3, 5×5, 7×7. Specifically, we implement reception fields of 5×5, 7×7 by utilizing convolution dilation based on 3×3 for efficiency. Table 1 shows the performance comparison of FMCB under different settings. Overall, equipping the backbone network with frame-level motion capturing block (FMCB), regardless of their settings, can significantly improve the performance (+6.7%-+6.9%) with neglectable external parameters and only 2% computation overhead. These results verify our claim that the motions in video activity involve multiple temporal granularities which may be ignored by models that explore video motions on only one temporal interval level. Particularly, we observe the increment of p from 2 to 4 which enlarges the spatial reception for prior and latter temporal frames from 3×3 to 5×5 slightly enhances the model with a performance improvement of 0.2% and further increasing p does not introduce performance improvement. This phenomenon indicates the raw frame with a bigger size than the feature maps in the latter stages of the network needs a bigger reception field than the 3×3 spatial kernel for the feature maps with smaller sizes. Since we utilize dilation convolution to implement the 5×5 spatial convolution, it doesn't introduce any external parameters and FLOPs, we fit p in FMCB as 4 for the following experiments.

As for the effectiveness of clip-level motion capturing block (CMCB), Table 2 shows its performance under different p. All results are obtained by densely inserting CMCB into the ResNet backbone equipped with FMCB under $p = 4$ which is denoted as "w/o CMCB" in Table 2. Similar to FMCBs, CMCBs under different settings of p consistently enhance the backbone with obvious performance improvements (+0.9%-+1.1%). However, increasing the reception field expansion slope could not further boost the effectiveness of CMCB, reflecting expand the reception field with slope $p = 2$ is enough to handle the movement of spatial clues in the feature maps which with smaller resolution size than raw frames. We fit p in CMCB as 2 for the following experiments.

Table 2: Performance comparison of different reception expansion slopes p for HgC in FMCB based on ResNet backbone equipped with FMCB (p=4) on the validation set of Something-Something V1.

method		top-1	top-5	#p	FLOPs
w/o CMCB		52.5	80.5	23.9M	33.6G
CMCB	**p=2**	**53.6**	81.4	24.1M	33.8G
	p=4	53.4	81.4	24.1M	33.8G
	p=6	**53.6**	81.8	24.1M	33.8G

Table 3: Performance comparison between HgC and rigid 1D temporal convolution in FMCB and CMCB on Something-Something V1. The p value is set as 4 and 2 in FMCB and CMCB respectively.

Motion level		acc.(%)		#P	FLOPs
FMCB	CMCB	top-1	top-5		
Conv1d	–	52.0	80.2	23.9M	33.6G
HgC		**52.5**	80.5	23.9M	33.6G
HgC	Conv1d	53.2	81.0	24.1M	33.8G
	HgC	**53.6**	81.4	24.1M	33.8G

After investigating the effectiveness of FMCB and CMCB under different settings, we further explore the effectiveness of HgCs in FMCB and CMCB. For a fair comparison, we replace the HgC in FMCB and CMCB with a rigid 1D temporal convolution and remain the other parts unchanged. Specifically, for HgCs in FMCB, we respectively integrate FMCBs which are implemented using HgC and 1D temporal convolution into ResNet50, and compare their performance on Something-Something V1. As shown in Table 3, using HgC in FMCB obtains a 0.5% performance improvement compared to 1D temporal convolution. This result verifies the capacity of our HgC in capturing micro motions across contiguous frames.

Based on backbone model equipped with FMCB, we replace the HgC in CMCBs using rigid 1D convolution in every bottleneck block of Res1-4 and make a comparison with original CMCBs. As shown in Table 3, insertion of CMCB using HgC introduces a larger improvement of 1.1% while the insertion of CMCB using rigid temporal 1D convolution only makes an improvement of 0.7%. Exhibiting stronger capability in both short-term and long-term motion capturing, our HgC is superior in spatio-temporal dynamics modeling than rigid 1D temporal convolution.

4.4 Performance Comparison

In this section, we compare H^2CN with state-of-the-art video networks. The result comparison follows the same protocol of using RGB frames as input unless otherwise specified. For the recent success of introducing transformer into action recognition, we compare H^2CN with both CNN-based and transformer-based architectures.

Table 4: Performance comparison with state-of-the-arts on Something-Something V1 and V2 datasets.

Method	Backbone	#Pretrain	Keyframes×Views	#P	FLOPs	V1		V2	
						Top-1	Top-5	Top-1	Top-5
I3D [3]	3DResNet-50	ImageNet	32×2	28.0M	153.0G×2	41.6	72.2	—	—
NLI3D [51]				35.3M	168.0G×2	44.4	76	—	—
NLI3D+GCN [52]				62.2M	303.0G×2	46.1	76.8	—	—
GST [34]	ResNet-50	ImageNet	16×1	21.0M	59.0G×1	48.6	77.9	62.6	87.9
TSM [30]	ResNet-50	ImageNet	16×1×2	23.9M	65.8G×1×2	48.4	78.1	63.1	88.2
SDA-TSM [44]			16×1×2	25.8M	67.8G×1×2	52.2	80.9	64.7	89.5
TIN [39]	ResNet-50	Kinetics	16×1	24.3M	67.0G×1	47	76.5	60.1	86.4
TEINet [31]	ResNet-50	ImageNet	16×1	30.4M	66.0G×1	49.9	—	62.1	—
TAM [33]	ResNet-50	ImageNet	16×1	25.6M	66.0G×1	47.6	77.7	62.5	87.6
TEA [27]	ResNet-50	ImageNet	16×30	24.5M	70.0G×30	52.3	81.9	—	—
STM [22]	ResNet-50	ImageNet	8×30	24.0M	33.3G×30	49.2	79.3	62.3	88.8
STM [22]			16×30	24.0M	66.5G×30	50.7	80.4	64.2	89.8
MoViNet-A3 [24]	—	—	50	5.3M	23.7G	—	—	64.1	88.8
TDN [49]	ResNet-50	ImageNet	(8+16)×1	26.1M	108.0G×1	55.1	82.9	67.0	89.5
SELFYNet [25]	ResNet-50	ImageNet	8×1	—	37.0G×1	52.5	80.8	64.5	89.4
SELFYNet [25]			16×1	—	77.0G×1	54.3	82.9	65.7	89.8
SELFYNet [25]			(8+16)×1	—	114.0G×1	55.8	83.9	67.4	91.0
TimeSformer-HR [2]	Transformer	Kinetics	16×3	121.4M	1703G×3	—	—	62.5	—
ViViT-L [1]			32×4	352.1M	903G×4	—	—	65.4	89.8
MViT-B [8]			64×3	36.6M	455G×3	—	—	67.7	90.9
Video-Swin-B [32]			16×3	88.8M	321G×3	—	—	**69.6**	**92.7**
H^{2}CN(ours)	ResNet-50	ImageNet	8×1	24.1M	33.8G×1	53.6	81.4	65.2	89.7
H^{2}CN(ours)			16×1	24.1M	67.6G×1	55.0	82.4	66.4	90.1
H^{2}CN(ours)			(8+16)×1	—	101.4G×1	**56.7**	**83.2**	**67.9**	**91.2**

Something-Something V1&V2. Since the recognition of videos in Something-Something V1&V2 relies heavily on time dynamic modeling, Something-Something V1&V2 are widely used to measure the motion capturing ability of video recognition methods. We report the top-1/top-5 precision of H^{2}CN with other SOTAs, model size and FLOPs are considered as well in Table 4. We compare our H^{2}CN with CNN-based architectures including classical methods such as I3D [3], GST [34], TSM [30] and most recent methods, such as TDN [49] and SELFYNet [25]. H^{2}CN achieves the highest Top-1 accuracies of 56.7% and 67.9& using (8+16) frames on Something-Something V1&V2, respectively. Compared to other CNN-based SOTAs, H^{2}CN outperforms them by obvious margins (0.9%-17.1% on V1 and 0.5%-6.8% on V2). These results demonstrate the capacity of H^{2}CN in capturing diverse activity motions. Compared to the more sophisticated Transformer-based methods like MViT [8] and Video-Swin transformer [32], the performance of our H^{2}CN is still competitive. What's more, H^{2}CN appears more efficient by consuming less computation and being with fewer parameters. H^{2}CN only requires 101.4G FLOPs, which is about 12.5 times cheaper than MViT-B (1365G FLOPs) and 8.5 times lower than Video-Swin-B (963G FLOPs).

Kinetics-400. Kinetics-400 has a very different temporal characteristic from Something-Something dataset, we conduct experiments on Kinetics-400 to verify the generality of the effectiveness of our H^{2}CN and HgC. On Kinetics-400, our H^{2}CN achieves 76.9% using 8 frames, while uses 16 frames, H^{2}CN reaches 77.9%. We further ensemble the 8-frames model and the 16-frames model and boost the performance to 78.7%. Compared with the classical video CNN backbones, spanning from 2D CNNs (e.g. TSN [50] and TSM [30]) to (decomposed) 3D CNNs (e.g. I3D [3] and R(2+1)D [47]), with large margin, H^{2}CN consistently outperforms those methods consuming less or similar computation. When compared with methods equipped by advanced motion capturing techniques (e.g. TDN [49], SELFYNet [25] and SlowFast [12]), our H^{2}CN still performs better at a similar or lower computation cost, except X3D-L [11] and MoViNet [24] which achieve efficiency through the reconstruction of the entire architecture of CNNs.

Table 5: Performance comparison on Kinetics-400.

Method	Backbone	Frames	GFLOPs	Top1	Top5
TSN [50]	InceptionV3	25	80×10	72.5	90.2
TSM [30]	ResNet50	16	65×30	74.7	91.4
I3D [3]	InceptionV1	64	—	72.1	90.3
R(2+1)D [47]	ResNet34	32	152×10	74.3	91.4
S3D-G [54]	InceptionV1	64×30	71.4×30	74.7	93.4
NL-I3D [51]	ResNet50	32	282×10	74.9	91.6
TEA [27]	ResNet50	16	70×30	76.1	92.5
TANet [33]	ResNet50	16	86×12	76.9	92.9
SmallBigNet [26]	ResNet50	8	57×30	76.3	92.5
SlowFast [12]	ResNet50	8+32	65.7×30	77.0	92.6
X3D-L [11]	—	16	24.8×30	77.5	92.9
MoViNet-A5 [24]	—	120	289	78.2	—
SELFYNet [25]	ResNet50	16	77×30	77.1	—
TDN [49]	ResNet50	8+16	108×30	78.4	93.6
H^{2}CN (Ours)	ResNet50	8	33.8×30	76.9	93.0
H^{2}CN (Ours)	ResNet50	16	67.6×30	77.9	93.3
H^{2}CN (Ours)	ResNet50	8+16	101.4×30	**78.7**	**93.6**

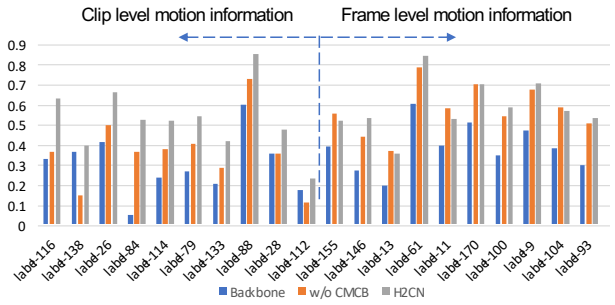


Figure 6: Per-category accuracy of ResNet backbone, H^{2}CN w/o CMCB, and H^{2}CN over 20 action categories. The 20 actions are improved most by CMCB (left of dotted line) or FMCB (right of dotted line).

4.5 Analysis and Visualization

We investigate the effectiveness of frame-level motion information and clip-level motion information by comparing the performance of full H^{2}CN, H^{2}CN w/o CMCB and ResNet backbone on different action categories of Something-Something V1, in particular, the performance margin between H^{2}CN, H^{2}CN w/o CMCB shows the effectiveness of clip-level motion information while the performance gap between H^{2}CN w/o CMCB and ResNet backbone demonstrates the capacity of frame-level motion information on a specific action category. Figure 6 shows the per-category performance comparison. The categories on the left of the dotted line are improved most by clip-level motion capturing compared to H^{2}CN w/o CMCB. The recognition of these categories needs more long-term motion information between clips than micro motion between contiguous frames, for example, the recognition of "*label-116: Putting something that can't roll onto a slanted surface, so it slides down*" relies on capturing long-term object movement. Contrary to categories on the left of the dotted line, the recognition of categories on the right of the dotted line relies more on micro motion information, such as the recognition of "*label-93: Pulling two ends of something so that it separates into two pieces*" only needs to focus on the moment when an object is divided into two pieces.

Based on the observation of recognition of different categories relies on motion information on different levels, we assume motion information on different levels works complementary. To verify this assumption, we calculate the performance margin between H^{2}CN w/o CMCB and ResNet backbone model which reflects the effectiveness of frame-level motion information as well as the performance margin between H^{2}CN and H^{2}CN w/o CMCB which reflects the effectiveness of clip-level motion information to investigate the relation of frame/clip-level motion information on all 174 categories of Something-Something V1. We respectively sort the performance margin between H^{2}CN and H^{2}CN w/o CMCB and the performance margin between H^{2}CN w/o and backbone then observe the trend of each other. For clear demonstration, we utilize third-order polynomial to smooth the trends. The trends are shown in Figure 7: as the effectiveness of clip-level motion information decreases, the recognition benefits more from frame-level motion information, and vice versa. This phenomenon verifies the motion information on different levels works in a complementary way.

Figure 7: Per-category accuracy margins between H^{2}CN w/o CMCB and ResNet backbone (orange), H^{2}CN and H^{2}CN w/o CMCB (blue) over all action categories on Something-Something V1. The trend of effectiveness of CMCB and FMCB shows CMCB and FMCB are complementary for motion modeling.

5 CONCLUSION AND FUTURE WORK

In this paper, we have presented a novel temporal convolution named as *Hourglass Convolution* (HgC) for its hourglass shaped reception field. With larger reception fields for prior and latter temporal frames/clips, HgC aggregates richer spatio-temporal semantics than rigid 1D temporal convolution. With the motion capturing capacity of HgC, we construct Hierarchical Hourglass Convolutional Network (H^{2}CN) for modeling both short-term (frame-level) and long-term (clip-level) motions in videos by separately applying HgC on consecutive frames and video clips. Being easily integrated into ResNet backbones, H^{2}CN results in SOTA performance on Something-Something V1&V2 of 56.7%/67.9% by only introducing 1%/3% extra parameters/FLOPs. SOTA performances are also obtained on Diving48 and EGTEA Gaze+ which have disparate data distributions, showing the generality of HgC and H^{2}CN.

This work represents an initial attempt to enhance the video motion modeling by designing a non-trivial temporal convolutional kernel rather than the widely used rigid 1D temporal convolution. Apart from the HgC presented in this paper, we would like to explore more flexible convolution kernel designs. In current version, HgC focuses on adjusting the spatial receptive field by considering the temporal offsets. So, how about resetting the temporal receptive field based on the spatial cues, for example, enlarging the temporal size as the spatial offset increases. Further more, all the above kernel designing regimes need to be pre-defined. Inspired by transformers that adaptively decide the importance of feature points, we seek for the adaptive HgC for learnable visual perception in the future. We hope our HgC and H^{2}CN provide some insights in the filed of video motion modeling.

ACKNOWLEDGMENTS

The work was supported in part by the National Key Research and Development Program of China (2020YFB1406703), by the National Natural Science Foundation of China (62101524), and by The University Synergy Innovation Program of Anhui Province (No. GXXT-2021-009).

REFERENCES

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. 2021. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6836–6846.

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. 2021. Is Space-Time Attention All You Need for Video Understanding?. In *International Conference on Machine Learning*. PMLR, 813–824.

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

[4] Cheng Chen, Zhenshan Tan, Qingrong Cheng, Xin Jiang, Qun Liu, Yudong Zhu, and Xiaodong Gu. 2022. UTC: A Unified Transformer with Inter-Task Contrastive Learning for Visual Dialog. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18103–18112.

[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. 2017. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*. 764–773.

[6] Jianing Deng, Li Wang, Shiliang Pu, and Cheng Zhuo. 2020. Spatio-Temporal Deformable Convolution for Compressed Video Quality Enhancement. *national conference on artificial intelligence* (2020).

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

[8] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. 2021. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6824–6835.

[9] Linxi Fan, Shyamal Buch, Guanzhi Wang, Ryan Cao, Yuke Zhu, Juan Carlos Niebles, and Li Fei-Fei. 2020. RubiksNet: Learnable 3D-Shift for Efficient Video Action Recognition. In *European Conference on Computer Vision*. Springer, 505–521.

[10] Chaowei Fang, Dingwen Zhang, Liang Wang, Yulun Zhang, Lechao Cheng, and Junwei Han. 2022. Cross-Modality High-Frequency Transformer for MR Image Super-Resolution. *arXiv preprint arXiv:2203.15314* (2022).

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

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

[13] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*. PMLR, 1243–1252.

[14] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. 2017. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE International Conference on Computer Vision*. 5842–5850.

[15] Yanbin Hao, Chong-Wah Ngo, and Bin Zhu. 2021. Learning to match anchor-target video pairs with dual attentional holographic networks. *IEEE Transactions on Image Processing* 30 (2021), 8130–8143.

[16] Yanbin Hao, Shuo Wang, Pei Cao, Xinjian Gao, Tong Xu, Jinmeng Wu, and Xiangnan He. 2022. Attention in Attention: Modeling Context Correlation for Efficient Video Classification. *IEEE Transactions on Circuits and Systems for Video Technology* (2022).

[17] Yanbin Hao, Hao Zhang, Chong-Wah Ngo, and Xiangnan He. 2022. Group Contextualization for Video Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 928–938.

[18] Yanbin Hao, Hao Zhang, Chong-Wah Ngo, Qiang Liu, and Xiaojun Hu. 2020. Compact Bilinear Augmented Query Structured Attention for Sport Highlights Classification. In *Proceedings of the 28th ACM International Conference on Multimedia*. 628–636.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[20] Berthold KP Horn and Brian G Schunck. 1981. Determining optical flow. *Artificial intelligence* 17, 1-3 (1981), 185–203.

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

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

[23] Manjin Kim, Heeseung Kwon, Chunyu Wang, Suha Kwak, and Minsu Cho. 2021. Relational Self-Attention: What's Missing in Attention for Video Understanding. *Advances in Neural Information Processing Systems* 34 (2021).

[24] Dan Kondratyuk, Liangzhe Yuan, Yandong Li, Li Zhang, Mingxing Tan, Matthew Brown, and Boqing Gong. 2021. MoViNets: Mobile Video Networks for Efficient Video Recognition. *computer vision and pattern recognition* (2021).

[25] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. 2021. Learning self-similarity in space and time as generalized motion for video action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 13065–13075.

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

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

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

[29] Yin Li, Miao Liu, and James M Rehg. 2018. 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)*. 619–635.

[30] Ji Lin, Chuang Gan, and Song Han. 2019. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7083–7093.

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

[32] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. 2021. Video Swin Transformer. *arXiv preprint arXiv:2106.13230* (2021).

[33] Zhaoyang Liu, Limin Wang, Wayne Wu, Chen Qian, and Tong Lu. 2020. TAM: Temporal Adaptive Module for Video Recognition. *arXiv preprint arXiv:2005.06803* (2020).

[34] Chenxu Luo and Alan L Yuille. 2019. Grouped spatial-temporal aggregation for efficient action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5512–5521.

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

[36] Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and João F Henriques. 2021. Keeping your eye on the ball: Trajectory attention in video transformers. *Advances in Neural Information Processing Systems* 34 (2021).

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

[38] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2016. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *International Journal of Computer Vision* (2016).

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

[40] Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems* 27 (2014).

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

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

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

[44] Yi Tan, Yanbin Hao, Xiangnan He, Yinwei Wei, and Xun Yang. 2021. Selective Dependency Aggregation for Action Classification. In *Proceedings of the 29th ACM International Conference on Multimedia*. 592–601.

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

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

[47] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. 2018. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 6450–6459.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS*.

[49] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. 2021. Tdn: Temporal difference networks for efficient action recognition. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition. 1895–1904.

[50] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. 2016. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*. Springer, 20–36.

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

[52] Xiaolong Wang and Abhinav Gupta. 2018. Videos as space-time region graphs. In *Proceedings of the European conference on computer vision (ECCV)*. 399–417.

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

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

[55] Hao Zhang, Yanbin Hao, and Chong-Wah Ngo. 2021. Token Shift Transformer for Video Classification. *acm multimedia* (2021).

[56] Hao Zhang, Yanbin Hao, and Chong-Wah Ngo. 2021. Token shift transformer for video classification. In *Proceedings of the 29th ACM International Conference on Multimedia*. 917–925.

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

[58] Shiwen Zhang, Sheng Guo, Weilin Huang, Matthew R Scott, and Limin Wang. 2019. V4D: 4D Convolutional Neural Networks for Video-level Representation Learning. In *International Conference on Learning Representations*.

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

[60] Andrew Zisserman, Joao Carreira, Karen Simonyan, Will Kay, Brian Hu Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, and Mustafa Suleyman. 2017. The Kinetics Human Action Video Dataset. *arXiv: Computer Vision and Pattern Recognition* (2017).

A PERFORMANCE COMPARISON ON DIVING48 [28] AND EGTEA GAZE+ [29]

Diving48. Diving48 is also a temporally-heavy dataset for the recognition of a diving pose relies on the temporal aggregation of sub-poses in the dive sequences. Since the new version Diving48 has revised for wrong labels, we re-train the CNN-based baselines, i.e. TSN [50] and TSM [30] using 16 frames and test them using 1 center crop as H^{2}CN. Table 6 shows the performance comparison. Compared with CNN baselines, our H^{2}CN achieves the best performance of 87.0%. What's more, our H^{2}CN performs better than the best Transformer-based method VIMPAC (85.5%).

Table 6: Performance comparison on the updated Diving48 dataset using the official train/validation split V2.

Method	Backbone	#Frame	Top-1
SlowFast,16×8 from [2]	ResNet101	64+16	77.6
TSN	ResNet50	16	79.0
TSM	ResNet50	16	83.2
SDA [44]	ResNet50	8	80.2
RSANet-R50 [23]	ResNet50	16	84.2
TimeSformer-HR [2]	Transformer	16	78.0
TimeSformer-L [2]	Transformer	96	81.0
VIMPAC [43]	Transformer	32	85.5
H^{2}CN(ours)	ResNet50	16	**87.0**

Table 7: Performance comparison on EGTEA Gaze+ dataset using the official train/validation split 1/2/3.

Method	Backbone	#Frame	Split1	Split2	Split3
I3D-2stream [29]	ResNet34	24	55.8	53.1	53.6
R34-2stream [42]	ResNet34	25	62.2	61.5	58.6
TSM [59]	ResNet50	8	63.5	—	—
SAP [53]	ResNet50	64	64.1	62.1	**62.0**
Vit(video) [7]	Transformer	8	62.6	—	—
TokShift [56]	Transformer	8	64.8	—	—
H^{2}CN	ResNet50	8	**66.2**	**63.9**	60.5

EGTEA Gaze+. Table 7 shows the performance of different methods on the first-vision EGATEA Gaze+ dataset. We compare our H^{2}CN firstly with methods which utilize 2 stream architecture [29, 42], although equipped with explicit motion information captured by optical flow, our H^{2}CN outperforms these methods with obvious margins on all three splits. Facing the specifically designed method for egocentric action recognition [53], our H^{2}CN outperforms SAP [53] on the split1 and split2 significantly with less data consumption. Then, we compare H^{2}CN with recent transformer-based methods. Our H^{2}CN significantly outperforms these methods, indicating the effectiveness of our H^{2}CN.

B SPATIO-TEMPORAL RESPONSE OF H^{2}CN

Figure 8 shows some examples of visualization results of spatio-temporal features of different models, including the backbone TSM, H^{2}CN w/o CMCB, and H^{2}CN. Specifically, heatmaps of their spatial features are computed by overlaying Grad-CAMs [38] of the Res4 feature on the input keyframes. Higher magnitudes in these heatmaps indicate higher feature responses for the regions. Since our H^{2}CN additionally has the temporal attention, we also visualize the keyframes with high attention weights using a red box as shown in the last row of each subfigure. The temporal attention is calculated from the last bottleneck of the Res4 stage. We use this visualization to clearly demonstrate the capture for both frame-level and clip-level motions of our H^{2}CN.

In general, with the receptive field expansion of HgC, H^{2}CN and H^{2}CN w/o CMCB demonstrate stronger dynamic perception in a larger area as the Grad-CAMs of H^{2}CN and H^{2}CN w/o CMCB attend to broader areas compared to the backbone. Particularly, in the recognition of the action "*Moving something across a surface until it falls down*" as shown in Figure 8 (a), the backbone model fails to focus on the interaction of "*hand*" and "*bottle cap*" over the whole video, resulting in an error category. As contrasts, H^{2}CN and H^{2}CN w/o CMCB manage to locate the moving interactions of "*hand*" and "*bottle cap*" and predict the correct category with the help of frame-level motion information. In addition, CMCB in H^{2}CN successfully pick the core keyframes (e.g., 6-8th keyframes) which contain the fallen "*bottle cap*" to confirm the prediction. As for the recognition of "*Rolling something on a flat surface*" in which the capturing of long-term movement plays a role of importance in Figure 8 (b), although frame-level motion information helps H^{2}CN w/o CMCB to locate the important motion areas more accurately than the backbone model, H^{2}CN w/o CMCB still fails to give the correct prediction for the lack of ability of modeling long-term movement. Equipped with long-term movement preception provided by CMCB which accurately attends to the keyframes when large object movement takes place (e.g. the 2-5th keyframes), H^{2}CN correctly recognizes the action.

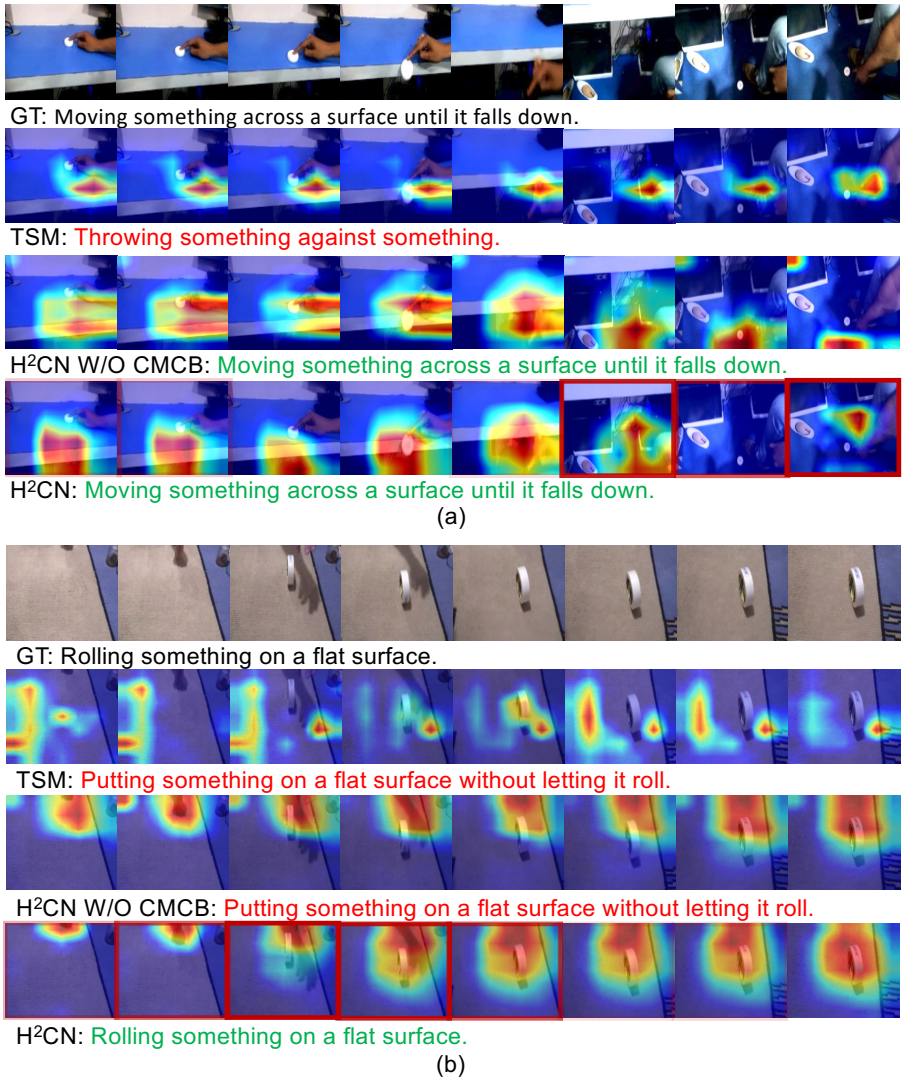


Figure 8: Spatio-temporal response of H^{2}CN. We compare H^{2}CN with the backbone model and H^{2}CN W/O CMCB.