Video-ception Network: Towards Multi-Scale Efficient Asymmetric Spatial-Temporal Interactions

Yuan Tian, Guangtao Zhai, and Zhiyong Gao

Abstract—Previous video modeling methods leverage the cubic 3D convolution filters or its decomposed variants to exploit the motion cues for precise action recognition, which tend to be performed on the video features along the temporal and spatial axes symmetrically. This brings the hypothesis implicitly that the actions are recognized from the cubic voxel level and neglects the essential spatial-temporal shape diversity across different actions. In this paper, we propose a novel video representing method that fuses the features spatially and temporally in an asymmetric way to model action atomics spanning multi-scale spatial-temporal scales. To permit the feature fusion procedure efficiently and effectively, we also design the optimized feature interaction layer, which covers most feature fusion techniques as special case of it, e.g., channel shuffling and channel concatenating. We instantiate our method as a plug-and-play block, termed Multi-Scale Efficient Asymmetric Spatial-Temporal Block. Our method can easily adapt the traditional 2D CNNs to the video understanding tasks such as action recognition. We verify our method on several most recent large-scale video datasets requiring strong temporal reasoning or appearance discriminating, e.g., Something-to-Something v1, Kinetics and Diving48, demonstrate the new state-of-the-art results without bells and whistles.

Index Terms—Action recognition, 2D CNN, asymmetric spatial-temporal modeling, temporal modeling, efficient CNN.

I. INTRODUCTION

VIDEO action recognition is an open challenging problem and has draw much attentions in both computer vision research community and industry by reason of its fundamental for tremendous applications, e.g., anomaly events analysis, human behavior monitoring, video surveillance, to name a few. Inspired by the breakthrough brought by CNNs on still image recognition task, i.e., the classification performance surpassing the human on ImageNet [3], the recent state-of-the-art video recognition methods also leverage the CNNs enhanced with temporal modeling ability for spatial-temporal modeling.

There are three promising temporal modeling strategies studied extensively. The first one is early temporal fusion where the temporal relation between consecutive frames are encoded before sending to the CNNs. For example, the two-stream architectures [4] takes optical flow map modalities as 2D CNN input. Dynamic image network [5] propose to compress a video clip to a novel single RGB dynamic image. The second one is late temporal fusion where the video frames are first encoded by 2D CNN and then fused. For instance, TSN [6] exploits several temporal consensus functions and achieves strong results. Timeception [7] leverages multi-scale temporal convolutions for long-range temporal modeling. The above two ways mostly model the spatial information and the temporal relations separately and may show instable performance across different datasets. Thus, the researchers seek for the 3D convolution as the third way to model the spatial appearance and the temporal motion simultaneously, which are widely adopted in the most recent video recognition architectures, e.g., C3D network [8], 3D-ResNet [9], R(2+1)D CNNs [10][11] and Slowfast networks [12]. The 3D convolution has several advantages for video recognition: (1) The 3 dimensional kernels couple the appearance and temporal
dynamics modeling, which is naturally powerful for spatial-temporal classification tasks such as action recognition task. (2) Expanding the 2D kernels into 3D kernels [13][14] converts the 2D CNNs to 3D CNNs immediately while leveraging the architecture designs [15][16] on ImageNet [3] and even their parameters.

However, the 3D CNNs show surprisingly inferior performances compared with the methods aggregating the 2D CNN results of each frame [17] or stacking other temporal reasoning modules [18][19][20][21] on top of 2D CNNs. This phenomenon can be demonstrated both quantitatively and intuitively. Quantitatively, on a appearance and scene biased action dataset, i.e., Kinetics, 3D-ResNet101 [14] (62.8%) is obviously surpassed by its 2D counterpart TSN [17] with BNInception (73.9%), which simply averages the classification results of each frames. On a temporal motion biased action dataset, i.e., Something-Something v1, 13D [22] with 3D-ResNet50 backbone (41.6%) achieves much inferior performance compared with TSM [19] with 2D-ResNet50 (49.7%) even with a much heavier computation cost (306 GFLOPs vs. 98 GFLOPs), which is a zero-parameter temporal reasoning module. Intuitively, as shown in Fig. 1, videos require diverse spatial or temporal receptive fields to be classified accurately, which can not be guaranteed by stacking few ordinary 3D convolutional kernels of size $3 \times 3 \times 3$.

Motivated by the issues above, we propose to modeling the spatial-temporal correlation between appearance and motion in an asymmetric manner, instead of the symmetric kernel of 3D convolution (usually $3 \times 3 \times 3$). More specifically, we propose the multi-scale efficient asymmetric spatial-temporal block (MS-EAST-Block) to first exploit the spatial cue with 2D spatial filters of different size and then aggregate the rich spatial features with 1D temporal filters spanning long-short ranges. By stacking the block, our method can model the actions of rich spatial-temporal scales in an hierarchical manner.

To keep our method extremely efficient, our method follows several efficient CNN architecture designing practices. (1) Partial channel transformation. GST [18] studies that the features can be divided into two channel groups, i.e., the static appearance and the dynamic motion. The motion group only occupies a relative small proportion. ResNext [23] also find the heavy redundancy in the vanilla ResNet [16]. Therefore, the MS-EAST-Block is also only performed on a small proportion of the feature channels (1/4 or 1/8). (2) Group convolution. The MS-EAST-Block performs spatial or temporal modeling separately in each group for each kernel size and rely on the channel interaction operation to fuse the multi-scale features. (3) Efficient feature interaction strategy. Previous methods [24][23] argue that the channel shuffling operation can perform channel interaction more efficiently and is zero-parameter. Inspired by them, we proposed the novel Optimized Feature Interaction Layer to fuse the spatial and temporal features in isolated groups, which covers most feature fusion techniques as special case of it. In the experiment part, we demonstrate that the proposed layer achieves conspicuous performance improvement while only introducing marginal learnable parameters and computation cost.

Through our extensive experiments on several large-scale action recognition datasets, i.e., Something-Something v1 [1], Kinetics [13] and Diving48 [2], we demonstrate that our method compares favorably with widely-used 3D CNNs and other light-weight temporal reasoning modules for video modeling, and achieves the new state-of-the-art results. We summarize our contributions as follows:

- We propose a novel simple efficient yet effective spatial-temporal representation method that fuses the features spatially and temporally in an asymmetric way to model action atomsics spanning various spatial-temporal scales, instantiated as a plug-and-play Multi-Scale Efficient Asymmetric Spatial-Temporal Block.
- We perform an extensive ablation analysis of the proposed method to study its effectiveness in video action recognition and also show the trade-offs between the computation cost and the performance following different designing practices.
- We achieve state-of-the-art or competitive results on several large scale video datasets with comparable parameters and FLOPs compared to existing approaches.

II. RELATED WORKS

A. Deep Video Recognition

Early works tend to fine-tune the 2D CNN networks on video dataset while relying on the optical flow input modality or post processing to capture the temporal information. Two-stream CNNs [25][4] takes in the RGB input (spatial stream) and the optical flow input (temporal stream) respectively. Simonyan et al. [25] first proposed the original framework. Feichtenhofer et al.[4] then studied several feature-level fusion strategies between the two-streams. Later, temporal segment network (TSN) [17] proposes a new spare frame sampling strategy and uses temporal consensus functions to aggregate features of each frame. These methods achieve favorable results on the small video datasets, e.g., UCF101 [26] and HMDB51 [27]. However, the limited parameter number and the unlearnable temporal modeling of the methods above may humble the performances on large scale datasets, e.g., Kinetics [13].

To learn the temporal evolution along with the spatial information simultaneously, 3D networks, e.g., C3D network [8], 13D [13], 3D-ResNet [9], R(2+1)D CNNs [10][11] and Slowfast networks [12], recently have gained much attention as another research line. C3D [8] network is the first 3D network with only few layers. However, it has a huge number of parameters and is relatively hard to train. 13D [13], which is abbreviated from inflated 3D networks, propose to inflate the 2D CNNs [15][16] pretrained on ImageNet [3] by copying weights and achieves promising results on Kinetics [13] dataset. 3D-ResNet [9] systematically evaluates several popular inflated structures and find that inflated 2D-ResNet inherits the advantages of ResNet [16], i.e., easy to converge and consume low computation cost. R(2+1)D [10] decomposes the 3D convolution into a 2D convolution followed by 1D convolution and learns discriminative enough features for action recognition. Slowfast networks [12] involves a slow
pathway to capture spatial semantics at low frame rate and a fast pathway to capture motion at fine temporal resolution. We value the merits of modeling spatial and temporal features separately.

B. Multi-scale CNN Architectures

CNNs are naturally equipped with multi scale feature representation ability due to the hierarchical stacking of convolution kernels. The VGGNet [28] propose to enlarge the receptive field by stacking more layers of small kernels than directly using large kernels, through which the VGGNet learns more powerful multi-scale representation than AlexNet [29]. More recently, modern CNN architectures design the multi-scale branches explicitly. The GoogLeNet [30] utilizes diverse filters with different kernel sizes in parallel to encode the multi-scale feature in each branch. However, the feature diversity is often limited by the computational constraints due to its limited parameter efficiency. Latter, the Inception Nets [31][15] propose to utilize more small filters in each branch of the parallel branches in the GoogLeNet [30] to further expand the receptive field. ResNet [16] allows different combinations of convolutional operators by short connection and obtains much deep network structure to achieve the more bigger receptive field.

Segmentation problems usually rely on large receptive field to capture the global contexts for finer results. Max pooling and dilation convolution (also named atrous convolution) [32] are widely adopted upon the backbone networks for this goal. Pyramid scene parsing network (PSPNet) [33] harvests different sub-region representations by concatenating the multi-level max-pooled features. DeepLab [34] proposes atrous spatial pyramid pooling (ASPP), where multi-atrous convolution layers with different rates capture multi-scale feature representations in parallel.

C. Efficient Neural Network Designing

State-of-the-art CNN architectures [16][23][35][36] discover that the redundancy in feature maps is an important characteristic. ResNet [16] propose the bottleneck design in each residual block, where the $1 \times 1$ layers are responsible for reducing and then increasing dimensions, leaving the interleaved $3 \times 3$ performed on a smaller input/output dimensions.

Grouped convolution and depistive convolution are then widely utilized due to their low computational cost. Grouped convolution is first introduced in AlexNet [29] for the purpose of distributing the model over multi GPUs and then widely used in later network architectures, e.g., ResNeXts [23], for efficient computing. Depistive convolution is the special case of grouped convolutions, where the feature channel of each group is single. Recent compact models running on mobile platforms such as MobileNetV2 [35] and ShuffleNet [24][37] leverage the depistive convolution extensively and achieve effective results. Particularly, ShuffleNet [24] propose a novel channel shuffling operation for fusing the features produced by different group convolution filters. This operation is more efficient and hardware-friendly then the ordinary $1 \times 1$ convolution.

III. APPROACH

In this section, we first adapt some 2D CNN architectures to their $(2+1)$D analogy for enlarging the receptive fields and extracting rich multi-scale features. However, the receptive fields are spatial-temporal symmetric. To exploit the spatial-temporal asymmetry existed widely in the real-word video actions, we propose the Optimized Feature Interaction Layer for enhancing the asymmetric interactive richness of spatial-temporal feature streams and then utilized this layer to build the Multi-Scale Efficient Asymmetric Spatial-Temporal Block. Finally, we develop a new efficient video modeling network upon the block with an extremely efficient architecture and high performance.

A. Multi-Scale Spatial-Temporal Modeling Architecture

In video recognition field, recent works such as R(2+1)D [10] and P3D [11] propose to decouple the 3D spatial-temporal filters of size $3 \times 3 \times 3$ into a $1 \times 3 \times 3$ convolutional filters acting as a 1D temporal filter. This decomposition reduces the computation cost and introduces more non-linearity to the whole architectures. Thus, we adopt the $(2+1)$D convolution instead of 3D convolution in our method.

Deep convolution neural networks, e.g., VGGNet [28], often stacks a large number of small convolution filters in an hierarchical manner for implicit multi-scale image representation learning. Following them, we propose the RF-L structure for capturing video features within Larger Receptive Field, as shown in Fig. ?? (a). Latter works such as InceptionNet [31] explicitly extract multi-scale features with isolated branches, each of which are composed of convolution kernels of different sizes. Group convolution has been widely used in light-weight CNN designing, for example, ResNext [23], ShuffleNet [24], channel-separated convolutional networks (CSN) [38], Inception networks [31], to name a few, which reduces the computation cost by several times. We adapt the Inception architecture [31][15] and propose the efficient RF-L-Inception structure for video modeling, as shown in Fig. 2 (b).
We propose to quantify the asymmetric spatial-temporal interaction of the structures above as the path numbers from one spatial feature channel to one temporal feature channel, where the sizes of the kernels for producing the spatial feature and the temporal feature are not equal. Given the output channel groups \( S = \{S_1, S_2, ..., S_G\} \) of the spatial convolutions and the input channel groups \( T = \{T_1, T_2, ..., T_G\} \) of the temporal convolutions, where \( G \) is the spatial or temporal group numbers\(^1\), we define the interaction between the \( S_i \) and \( T_j \) as the number of the channels flowed from \( S_i \) to \( T_j \), denoted as \( N_{S_i \rightarrow T_j} \). The asymmetric interaction is formulated as:

\[
interactions = \sum_{i=0}^{G} \sum_{j=0}^{G} N_{S_i \rightarrow T_j}, i \neq j.
\] (1)

Obviously, \( interactions \) is 0 for the structures above, indicating their lack of asymmetric spatial-temporal modeling capability.

B. Optimized Feature Interaction Layer

In this section, we propose a novel optimized feature interaction layer to devise the efficient asymmetrical spatial and temporal pathways, as shown in Fig. 3.

Given the features \( X = X_1 \oplus X_2 \oplus \ldots \oplus X_G \) produced by spatial convolutions, where \( G \) is the group number. Now, we assume there exists a transformation matrix \( F \in \mathbb{R}^{G \times G} \) performing on \( X \) to produce \( Y = Y_1 \oplus Y_2 \oplus \ldots \oplus Y_G \). \( Y \) will be convolved with \( G \) temporal convolution kernels of different sizes. The channel number of each group of \( X \) and \( Y \) are all denoted as \( C \). In the following parts, we omit the domain of \( i \in [1, G] \) and \( j \in [1, G] \). We formulate the above as:

\[
Y = F \times X,
\] (2)

\[
\begin{bmatrix}
Y_1 \\
\vdots \\
Y_G
\end{bmatrix} =
\begin{bmatrix}
W_{11} & \ldots & W_{1G} \\
\vdots & \ddots & \vdots \\
W_{G1} & \ldots & W_{GG}
\end{bmatrix}
\begin{bmatrix}
X_1 \\
\vdots \\
X_G
\end{bmatrix},
\] (3)

where the \( Y_i \) is the linear weighting sum of \( \{X_1, X_2, \ldots, X_G\} \):

\[
Y_i = W_{i1} \cdot X_1 + W_{i2} \cdot X_2 + \ldots + W_{iG} \cdot X_G,
\] (4)

We also assume that each group \( X_i \) of \( X \) is also sub-grouped into \( G \) sub-groups: \( X_i = X_{i1} \oplus X_{i2} \oplus \ldots \oplus X_{iG} \). Then, the multiplications between the weighting parameter \( W_{ij} \) and the feature \( X_i \) in each group also follows the group decomposition calculation rules:

\[
W_{ij} \times X_i = (W_{ij1} \times X_{i1}) \oplus (W_{ij2} \times X_{i2}) \oplus \ldots \oplus (W_{ijG} \times X_{iG}),
\] (5)

Optimization goal:

1) Asymmetric Spatial-Temporal Interaction Richness. Each group \((Y_j)\) of \( Y \) should contain the information partially flowed from every group of \( X \). From Eq. 4, we have:

\[
|W_{ij}| > 0,
\] (6)

\(^1\)We assume the spatial and the temporal features are divided into groups of equal number for simplicity.

This term facilitates higher intra-group interaction richness of \( X \) taken input by the subsequent temporal convolutions.

2) High network capacity. Previous works \([35][39][40]\) demonstrate that the wider networks lead to better performance and the width is even more important than the depth of networks in some architectures \([39]\). Inspired by them, we assume the network capacity is positive about the number of effective feature groups. To ensure the effectiveness of every group, each feature group \( Y_i \) of \( Y \) should be orthogonal with the other groups, then:

\[
Y_i \cdot Y_j = 0, i \neq j,
\] (7)

Through combining Eq. 7 and Eq. 4, we have:

\[
|W_{i1}, \ldots, W_{iG}| \cdot |W_{j1}, \ldots, W_{jG}|^T \rightarrow 0, i \neq j,
\] (8)

This term facilitates the lower inter-group similarity of \( Y \) and thus leads to the wider network of higher capacity.

3) Sparse Transformation Matrix.

\[
|F| = \sum_{i=1}^{G} \sum_{j=1}^{G} |W_{ij}| \rightarrow 0,
\] (9)

This term penalizes the large weights in the transformation matrix \( F \) for numerical stability.

In practice, \( F \) can be implemented as a differential linear transformation layer and learned through end-to-end training simultaneously with other differential components while regularized by the regular terms (Eq. 6 8 9) above.

Discussions:

1) vs. Channel grouping operation. If relaxing the fullness regularization formulated by Eq. 6, one particular solution is

\[
\{ W_{ij}(k) = 1, i = j, k = 0, W_{ij}(k) = 0, others. \}
\] (10)
The operation of the proposed layer is degenerated to the channel grouping operation where the each output group feature is identical to that of the input group. This results in interaction = 0.

(2) vs. Dropout operation [41]. If relaxing the network capacity regularization formulated by Eq. 8, one particular solution is

\[
\begin{align*}
W_{ij}(k) &= 1, k = 0, \\
W_{ij}(k) &= 0, \text{others}.
\end{align*}
\]

(11)

\[
Y_i = X_{1i} \oplus X_{2i} \oplus ... \oplus X_{Gi}.
\]

(12)

The operation of the proposed layer is degenerated to the Dropout [41] followed by a concatenating operation where the each output group feature is the summing of the first \(\frac{1}{G}\) part of every input group. This indeed improve the interaction to \(G \cdot C\). However, all the output groups are the same feature, reducing the network capacity by \(\frac{1}{G}\) times.

(3) vs. Channel shuffle operation [24]. This operation can be implemented by Eq. 3 conditioned with

\[
\begin{align*}
W_{ij}(k) &= 1, k = i, \\
W_{ij}(k) &= 0, \text{others}.
\end{align*}
\]

(13)

This improve the interaction to \(G \cdot C\) and also retain the original network capacity. Moreover, we surprisingly find that this solution fulfills both the three regularizations above.

C. Multi-Scale Efficient Asymmetric Spatial-Temporal Block

In this section, we combine the RF-L-Inception structure in Fig. 2 and the optimized feature interaction layer in Fig. 3 to build the Multi-Scale Efficient Asymmetric Spatial-Temporal Block for extremely efficient multi-scale spatial-temporal video feature extraction, as shown in Fig. 4.

Spatial-temporal feature decoupling. Decomposing the 3D filter into 2D filter followed by 1D filter has gained big success in R(2+1)D Net [10] and its improved variants such as video correlation networks [42]. Unlike them, the MS-EAST block performs multi-scale spatial and temporal modeling efficiently compared to the single scale spatial-temporal feature extracting proficiency of R(2+1)D Net [10] in each layer.

Bottleneck design. GhostNet [36] demonstrates there exist many similar pairs of feature maps in each layer of ResNet50. Group spatial-temporal aggregation (GST) [18] also find that only performing spatial-temporal modeling on a small proportion of the original 2D CNN features can achieve more accurate action recognition then its 3D counterparts. Inspired by them, we adaptively select a subset (e.g., \(\frac{1}{4}\) or \(\frac{1}{2}\)) of the input features, then perform multi-scale asymmetric spatial-temporal modeling on the selected feature and restore the channel number by \(1 \times 1\) convolution operation.

Dilation convolution. Inspired by the utilizing of dilated convolutions for multi-scale context aggregation [32], we also leverage this technique to exponentially expand receptive fields of convolution operation without introducing extra computational costs or parameters. The dilated spatial and temporal convolutions are formulated respectively as:

\[
Y(x, y) = \sum_{i=0}^{W} \sum_{j=0}^{H} X(x + r \cdot i, y + r \cdot j)w(i, j),
\]

(14)

and

\[
Y(t) = \sum_{i=0}^{T} X(t + r \cdot i)w(i),
\]

(15)

where X and Y denote the input and output features separately, \(w\) denotes the learnable filter, \(r\) is the dilation rate, \(W, H\) and \(T\) is the spatial and temporal size of the input features.

Identity and max-pooling operations. We leverage the identity operation instead of the \(1 \times 1\) convolution in the spatial feature extracting stage for that the feature interaction layer and the following temporal modeling stage will both interacting the features of different channels. Moreover, we replace partial \(3 \times 3\) convolution with max-pooling operation of kernel size \(3 \times 3\) to preserve high-frequency part of some features, which is complementary to the low-frequency features extracted by spatial convolution operation. Additionally, the two strategies above reduce the number of parameters further and thus alleviate the risk of over-fitting.
The proposed MS-EAST-Block is flexible and can be easily integrated with most of the current networks stacked by 2D or 3D convolutions. More specifically, we adopt 2D-ResNet [16] as the backbone networks and insert the proposed MS-EAST-block between the residual blocks. We customize the spatial-temporal receptive filed by leveraging a different number of this block. The final prediction is a simple average pooling of each frame. The whole architecture above is termed as Video-ception network. We also conduct experiments to show if other more sophisticated late feature fusion method such as TRN [43] and ECO [44] can improve the performance further.

IV. EXPERIMENTS

A. Video Datasets

We evaluate our model on four large-scale video datasets that have rather different properties, corresponding to the distinct aspects of video recognition.

**Something-Something.** This dataset includes v1 [1] and v2 [45], which are two large scale crowd-sourcing video datasets for action recognition. There are totally about 110k (v1) and 220k (v2) videos for 174 fine-grained classes with diverse objects and scenes, focusing on humans performing pre-defined basic actions. The same action is performed with different objects (something) so that models are forced to understand the basic actions instead of recognizing the objects. Moreover, the required spatial-temporal receptive-field varies hugely across different fine-grained level actions as shown in Fig. 1 (a), which is very suitable for verifying the flexible spatial-temporal modeling ability of the proposed method. We mainly conduct ablation experiments and justify each component on Something-Something v1 dataset.

**Kinetics.** Kinetics [13] is a challenging human action recognition dataset, which contains 400 and 600 human action classes. The actions includes human-object interactions such as playing instruments, as well as human-human interactions such as shaking hands and hugging. Compared to the temporal reasoning required by the actions in Something-Something, the actions in this dataset mainly rely on the appearance of the objects and the background scenes to be discriminated. There are two versions of this dataset: untrimmed and trimmed. The untrimmed videos contain the whole video in which the activity is included in a short period of it. However, the trimmed videos contain the activity part only. We evaluate our models on the trimmed version to prove the high appearance network capacity and report the accuracy on the validation set. We conduct the experiments on Kinetics-400 [13] because there are many well known baseline methods benchmarked on this dataset.

**Diving48.** Diving48 [2] is a newly released dataset with more than 18K video clips for 48 unambiguous diving classes. This proves to be a challenging task for modern action recognition systems as dives may differ in three stages (takeoff, flight, entry) and thus require modeling of long-term temporal dynamics. Particularly, the actions can only be recognized by first discriminating the short-term action atomics and then reasoning the long-term dependence order between them, as shown in Fig. 1 (b). This requires multi-scale temporal modeling. Therefore, we conduct experiments on this dataset to verify the multi-scale spatial-temporal modeling ability of our method comprehensively. We report the accuracy on the official train/val split.

B. Implementation Detail

We implement our model in Pytorch [46]. We adopt ResNet50 [16] pretrained on Imagenet [3] as the backbone. The parameters within the MS-EAST-Blocks are randomly initialized. For the temporal dimension, we use the sparse sampling method described in TSN [6]. Specifically, the videos are first divided into several segments of equal duration, and then one snippet is randomly sampled from its corresponding segment. The snippets forms the clip input to the networks. For spatial dimension, the short-side of the input frames are resized to 256 and then cropped to 224 × 224. We do random cropping and flipping as data augmentation during training time. It’s worth to note that we do not perform horizontal flipping on some moving direction related action classes such as ”moving something from left to right”. We train the network with a batch-size of 72 on 6 NVIDIA GTX-2080Ti GPUs and optimize using SGD with an initial learning rate of 0.01 for about 40 epochs and decay it by a factor of 10 every 10 epochs. The total training epochs are about 70. The dropout ratio is set to be 0.3 as in [18]. During the inference time, we sample the middle frame in each segment and do center crop for each frame. We report the results of 1 crop unless specified. Noting that many state-of-the-art methods report their final performances with 5 or 10 crops, which enlarge the inference-time computation cost by 5 or 10 times. Moreover, We only use RGB modality as the input to our model, unlike two-stream networks [25][4] which use both RGB and optical-flow modalities.

C. Ablation Study

We conduct extensive ablation studies on the Something-Something V1 [1] dataset to demonstrate the effectiveness of every aspects of our method. All the MS-EAST-Blocks in this experiments are of receptive field $5 \times 5 \times 5$ and performed on the $\frac{1}{4}$ proportion of input features if no specified otherwise. To facilitate the training process in these experiments, we adopt a smaller input resolution: the short-side of the input frames are resized to 128 and then cropped to $112 \times 112$. The batch size is of 120. For the convenience of expression, the five blocks of ResNet-50 are denoted as conv1, conv2, conv3, conv4 and conv5 respectively.

For verifying the importance of multi-scale and asymmetric modeling of our method, we compare against the following baselines:

- **TSN model.** The vanilla TSN based on ResNet-50 model.
- **RF-S model.** The spatial-temporal modeling block in this model is implemented as $3 \times 3 \times 3$ (2+1)D convolutions, which only capture single-scale features with relative smaller receptive field.
- **RF-L model.** The spatial-temporal modeling block in this model is implemented as stacked two $3 \times 3 \times 3$ (2+1)D
TABLE I: Results of inserting different spatial-temporal modeling blocks to 2D ResNet-50 feature extraction network. † indicates that the results are reproduced under the input clips of $8 \times 112 \times 112$ by leveraging the codes open-resourced by the authors.

| Methods         | Backbone | Receptive field | Asymmetric | Multi-scale | Parameter number | Something-Something v1 |
|-----------------|----------|-----------------|------------|-------------|------------------|------------------------|
| TSN [6]†        | ResNet50 | -               | -          | -           | 23.87M           | 14.90 36.97           |
| TSM [19]†       | ResNet50 | 1 × 3           | ✓          | -           | 23.87M           | 42.11 71.01           |
| GST [18]†       | ResNet50 | 3 × 3           | -          | -           | 21.04M           | 42.36 71.42           |
| GST [18]†       | ResNet101| 3 × 3           | -          | -           | 37.52M           | 41.16 69.76           |
| RF-S            | ResNet50 | 3 × 3           | -          | -           | 27.57M           | 43.72 73.14           |
| RF-L            | ResNet50 | 5 × 5           | ✓          | -           | 27.35M           | 43.67 72.87           |
| RF-L-Inception  | ResNet50 | 5 × 5           | ✓          | ✓           | 27.56M           | 43.84 72.59           |
| RF-L-Inception-T| ResNet50 | 1 × 5           | ✓          | ✓           | 27.28M           | 44.23 72.87           |
| Ours            | ResNet50 | 5 × 5           | ✓          | ✓           | 27.55M           | 45.44 74.33           |
| Ours            | ResNet101| 5 × 5           | ✓          | ✓           | 27.56M           | 45.44 74.33           |
| Ours-1/16       | ResNet50 | 5 × 5           | ✓          | ✓           | 24.05M           | 44.00 72.65           |

TABLE II: Results of inserting different spatial-temporal modeling blocks to ResNet-50 feature extraction network. Larger Asymmetric interaction and spatial feature utilization bring more performance gain.

| Feature interaction methods | Asymmetric interaction | Spatial feature utilization | Parameters | Something-Something v1 |
|-----------------------------|------------------------|---------------------------|------------|------------------------|
| channel grouping            | 0                      | 1                         | 0          | 43.69 72.10            |
| dropout                     | 232                    | 0.25                      | 0          | 44.64 73.03            |
| channel shuffling           | 232                    | 1                         | 0          | 45.01 73.53            |
| Ours /wo Reg                | <232                   | <1                        | 0.086 M    | 45.22 73.93            |
| Ours                        | ~232                   | ~1                        | 0.086 M    | 45.44 74.33            |

TABLE III: Results of inserting MS-EAST-Block to ResNet-50 feature extraction network. The accuracy is consistently better, when more blocks are progressively inserted into the shallow layers.

| Usage of MS-EAST-Block (# layers) | Something-Something v1 |
|-----------------------------------|------------------------|
| none (0, baseline)               | 14.90 36.97            |
| res1 (1)                          | 34.24 61.84            |
| res1,2 (2)                        | 38.27 66.52            |
| res1,2,3 (3)                      | 41.13 69.90            |
| res1,2,3,4 (4)                    | **45.44** 74.33        |
| res1,2,3,4,5 (5)                  | 43.24 71.68            |
| res2,3,4,5 (4)                    | 43.88 72.24            |

TABLE IV: Results of different feature proportions for spatial-temporal modeling.

| Usage of MS-EAST-Block (# layers) | Something-Something v1 |
|-----------------------------------|------------------------|
| 0                                 | 14.90 36.97            |
| 1/2                               | 44.93 73.66            |
| 1/4                               | **45.44** 74.33        |
| 1/8                               | 44.83 72.81            |
| 1/16                              | 44.00 72.65            |

TABLE V: Detailed design of MS-EAST-Block. Max-pooling operation, inter ReLU strategy and dilation convolution both improve the performance.

- **RF-L-Inception model.** The spatial-temporal modeling block in this model is implemented as Inception-like (2+1)D convolution group as shown in Fig. 2 (b), which capture multi-scale features with larger receptive field. However, the spatial and temporal receptive field is symmetric.
- **RF-L-Inception-T model.** The spatial-temporal modeling block in this model is implemented as Inception-like 1D convolution group, which only capture multi-scale temporal features with larger receptive field. This model is utilized for comparison with other methods, e.g., TSM, only modeling along the temporal axis.

**Design**  
| Parameter number | Something-Something v1 |
|------------------|------------------------|
| Avg Pool         | 27.55M 45.42 73.98     |
| Max Pool         | 27.55M **45.44** 74.33 |
| Wo Max Pool      | 27.55M 44.19 73.45     |
| Max Pool         | 27.55M **45.44** 74.33 |
| Wo inter ReLU    | 27.55M 44.65 73.30     |
| Inter ReLU       | 27.55M **45.44** 74.33 |
| Wo dilation      | 27.95M **45.54** 73.09 |
| Dilation         | **27.55M** 45.44 74.33 |

**Usage of MS-EAST-Block (# layers)**

| Something-Something v1 |
|------------------------|
| 14.90 36.97            |
| res1 (1)               | 34.24 61.84            |
| res1,2 (2)             | 38.27 66.52            |
| res1,2,3 (3)           | 41.13 69.90            |
| res1,2,3,4 (4)         | **45.44** 74.33        |
| res1,2,3,4,5 (5)       | 43.24 71.68            |
| res2,3,4,5 (4)         | 43.88 72.24            |

**Usage of MS-EAST-Block (# layers)**

| Something-Something v1 |
|------------------------|
| 14.90 36.97            |
| 44.93 73.66            |
| **45.44** 74.33        |
| 44.83 72.81            |
| 44.00 72.65            |
To keep the parameter number of the baselines consistent with our method, we set the proportion of features flowing into spatial-temporal modeling block as 1/6, 1/8, 1/4 and 1/4 for RF-S, RF-L, RF-L-Inception and RF-L-Inception-T model respectively.

1) The importance of multi-scale asymmetric spatial-temporal modeling: All the spatial-temporal modeling blocks of the baselines and our model are inserted after the conv1, conv2, conv3 and conv4 blocks, following the residual bottleneck design. As shown in Tab. I, our method outperforms both other state-of-the-art methods and our baselines largely. This shows the asymmetric multi-scale spatial-temporal modeling with larger receptive field can mine the temporal reasoning information much better. Also, compared with the RF-L-Inception model, our method with asymmetric spatial-temporal modeling ability improves the performance significantly with almost no extra computation cost. It’s worth to mention that asymmetric modeling is far more effective than the multi-scale designing. Multi-scale learning only contributes marginal performance gain: 0.73% (from RF-S model to RF-L-Inception) whereas asymmetric modeling bring conspicuous improvement: 1.75% (from RF-L-Inception to our method).

Moreover, all of our baselines outperform TSM network obviously, which exchanges part of the channels along temporal dimension and invalidates this part of the features, indicating the importance of keeping the original feature structure of the 2D backbone CNNs. To eliminate the influence of the extra spatial modeling introduced by our method, we propose the RF-L-Inception-T baseline consists of only temporal convolutions. The performance comparison also prove the merits of multi-scale feature extraction paradigm.

Partially similar to our method, GST also converts some 2D spatial features to 3D spatial-temporal features while utilizing 3D convolutions of $3 \times 3 \times 3$ spatial-temporal receptive field. However, all of our baselines except for the RF-S model outperforms GST, proving the superiority of utilizing larger spatial-temporal receptive field.

To compare with TSM strictly, we set the spatial-temporal modeling proportion of our method as 1/16, resulting the comparable parameter number as them. Our method outperforms TSM by 1.89% in terms of Top1 accuracy although TSM perform temporal modeling on more feature maps (1/8 of the input feature maps).

2) The Superiority of Optimized Feature Interaction Layer: We conduct experiments to compare the proposed optimized feature interaction layer with other feature interaction methods, as shown in Tab. II. When utilizing plain channel grouping between the spatial and the temporal feature extractors, our method degenerates to the RF-L-Inception baseline and shows the most inferior performance. Channel shuffling operation have the richest asymmetric spatial-temporal interaction and the highest spatial feature utilization percentage. However, the performance is slightly worse than our method because the operation is unlearnable and thus can not select the interacting features automatically in a data-driven way.

3) Other designs: Where to insert the block? The proposed MS-EAST-Block can replace any identity path of the backbone network and convert partial spatial features to multi-scale spatial-temporal features immediately. In this section, we consider to evaluate the impact of inserting the MS-EAST-Blocks after different Resnet blocks, as shown in Tab. III. Accuracy steadily improves when more MS-EAST-Block are used and saturates when inserting after the res1, 2, 3 and 4 layers. Therefore, in all of our experiments, we insert four MS-EAST-Blocks into backbone network as above respectively if not specified otherwise. Interestingly, we get much inferior performance when trying to insert the block after the res 2, 3, 4 and 5 blocks as previous works [47]. This is intuitive because our block has larger receptive field and requires to be performed on feature maps of larger spatial resolution.

What proportion of features need to be converted into space-time features? Our method convert some spatial channels of original 2D CNN to spatial-temporal channels. As shown in Fig. IV, we achieve the best result with utilizing $\frac{1}{16}$ proportion of the original features. Too high proportion will reduce the spatial modeling capacity and corrupt the features of the pretrained models on the image tasks, whereas too low proportion leads to insufficient spatial-temporal modeling capacity. Moreover, we find only applying our method to $\frac{1}{16}$ features also outperforms the TSN baseline by over 29%, demonstrating that rich spatial information are much important even on the datasets with temporal reasoning annotations, e.g., Something-Something, while temporal information can be small proportion but necessary.

Are features learned by convolutions enough? Recent works on images [31] and videos [48] show that the pooling operations can improve the feature expression of convolutions manifestly. R(2+1)D network [10] also interprets partial performance gain to the extra linearity introduced by the ReLU operation inserted between the spatial convolutions and the temporal convolutions. Previous works argue that the dilation convolution can improve the performance by capturing long range relationships without introducing extra parameters. In this part, we conduct experiments to verify if these designs are complementary to our method. As shown in Tab. V, the pooling operation improves the performance significantly (1.25% Top1 accuracy) while our method is not sensitive to the specific implementation of the operation. Either average pooling or max pooling can achieve excellent performance. Max pooling has slight advantage over average pooling because the regions of focusing objects and the key frames related to action state changes are only a small proportion of the input video data. Also, we find that the extra non-linearity introduced by the intermediate ReLU operations between spatial convolution and temporal convolution also benefits the performance obviously, which is consistent with the conclusion from previous works [10]. Finally, we demonstrate that the dilated convolution outperform the ordinary convolution by 1.24% with less parameters in terms of Top5 accuracy, showing the better generalization ability.

D. State-of-the-Art Comparison

1) Something-V1: The recognition performance obtained by our method is compared with state-of-the-art approaches that just use RGB frames, as shown in Tab. VI. We also report
Fig. 5: The action classes with the highest improvement over GST [18] on Something-Something v1 dataset [1]. X-axis shows the percentage of corrected samples for each class. Y-axis labels are the corresponding action categories.

| Method       | Backbone         | Pre-training | #Frames | GFLOPs | Top-1 (%) | Top-5 (%) |
|--------------|------------------|--------------|---------|--------|-----------|-----------|
| I3D [13]     | 3D-ResNet-50     | Kinetics     | 32×3×2  | 153×3×2| 41.6      | 72.2      |
| Non-local [49]| 3D-ResNet-50     | Kinetics     | 32×3×2  | 168×3×2| 44.4      | 76.0      |
| GCN+Non-local [22] | 3D-ResNet-50 | Kinetics     | 32×3×2  | 303×3×2| 46.1      | 76.8      |
| ECO(En) [44] | BNInc + 3D-ResNet-18 | Kinetics | 92×1×1  | 267×1×1| 46.4      |           |
| ECO(En)+flow [44] | BNInc + 3D-ResNet-18 | Kinetics | 92×92   | NA     | 49.5      | -         |
| TSN [6]      | BN-Inception     | ImageNet     | 8       | 16     | 19.5      | -         |
| TSN [6]      | BN-Inception     | ImageNet     | 16      | 32.73  | 17.52     | -         |
| MultiScale TRN [43] | BN-Inception | ImageNet     | 8       | 16.37  | 34.44     | 63.2      |
| MultiScale TRN+flow [43] | BN-Inception | ImageNet     | 8       | -      | 42        | -         |
| R(2+1)D [7] | ResNet-34        | Sports-1M    | 32      | 152    | 45.7      | -         |
| S3D-G [50]   | InceptionV1      | ImageNet     | 64      | 71.38  | 48.2      | 78.7      |
| TSM [19]     | ResNet-50        | Kinetics     | 8       | 33     | 45.6      | 74.2      |
| TSM [19]     | ResNet-50        | Kinetics     | 16      | 65     | 47.2      | 77.1      |
| STM [21]     | ResNet-50        | ImageNet     | 8×3×10  | 33×3×10| 49.2      | 79.3      |
| STM [21]     | ResNet-50        | ImageNet     | 16×3×10 | 67×3×10| 50.7      | 80.4      |
| GST [18]     | ResNet-50        | ImageNet     | 16      | 59     | 48.6      | 77.9      |
| TEA [20]     | ResNet-50        | ImageNet     | 8×1×1   | 35×1×1 | 48.9      | 78.1      |
| TEA [20]     | ResNet-50        | ImageNet     | 16×1×1  | 70×1×1 | 51.9      | 80.3      |
| TEA [20]     | ResNet-50        | ImageNet     | 16×3×10 | 70×3×10| 52.3      | 81.9      |
| Ours         | ResNet-50        | ImageNet     | 8       | 36.19  | 49.78     | 78.37     |
| Ours         | ResNet-50        | ImageNet     | 16      | 72.38  | -         | -         |
| Ours         | ResNet-50        | ImageNet     | 8×16    | -      | -         | -         |
| Ours         | ResNet-50        | ImageNet     | 16×3×10 | -      | -         | -         |

TABLE VI: Comparison to state-of-the-art on Something-V1. – indicates the paper didn’t provide the results.

E. Visualization

In Fig. 5, we show the top 10 action classes that improved most by utilizing MS-EAST-Block instead of GST block upon the CNN backbone. From the figure, it can be seen that the MS-EAST-Block brings more multi-object interaction modeling and long-short term temporal reasoning abilities to the backbone 2D CNNs than GST and improve the recognition of some classes such as "Spilling something next to something" and "Pulling two ends of something so that it separates into two pieces". To give a better understanding of these performance improvements, we randomly select one video for each one class above and feed the video into our network and GST network based on ResNet-50 respectively. We extract the 8×14×14 convolution feature from res4 block and aggregate the feature maps along the channel dimension by averaging into the activation heatmaps. Finally, we overlay the heatmaps...
Fig. 6: Feature Visualization and prediction evolution. Compared to GST [18], our MS-EAST-Block can discover more semantically consistent RoI for actions as well as reduce noisy backgrounds for correct prediction. In our result, we also average the scores for adjacent same prediction results. Green bars show the correct prediction for the whole video clip. The predicted result with all class confidences lower than 0.4 is considered as background. Our method also detect actions accurately although trained only with classification label.
on the original images for clear comparison.

Fig. 6 clearly demonstrate that our MS-EAST-Block can discover more semantically consistent RoI for actions as well as reduce noisy backgrounds for correct prediction. Moreover, the feature activation heatmaps of our method are more concentrated spatially. The results are originated from two aspects: (1) Our optimized interaction layer select more effective features by learning end-to-end while GST always leverages the first group features. (2) max-pooling branch in our block denoises the features. More concretely, for the video from class "Spilling something next to something", our method focus on the center region of surface from the first frame even though the water is spilled after several frames. This illustrates that our larger receptive design enables the interactions of long-term action cues and also largely improve other action classes needing to be inferred based on the state changes of all frames, e.g., "Pushing something from left to right".

The stronger feature representation of our method provide more accurate fine-grained action classification proficiency. For example, our method distinguish the similar action pairs such as "spilling" vs. "pouring" and "pulling" vs. "tearing" clearly while GST confuses them.

We also show the temporal evolution process below the heatmap in Fig. 6. Interestingly, our method also detect the start and end of actions accurately although trained only with classification labels. In the first example, when the water begin to spill on the surface, the action state is changed from Background to Spilling something next to something. Then, the predicted state is also evolved correspondingly when the spilled water begins to spread on the surface. This implies that our method can be extended to a weakly supervised action detection method in the future.

V. CONCLUSION

We proposed the multi-scale efficient asymmetric spatial-temporal block for more efficient and effective video modeling. Based on this block, we propose the video-ception network. We performed an extensive evaluation to study its effectiveness in video action recognition, achieving state-of-the-art results on Something Something-V1 datasets. We also demonstrate some visualization results for more intuitive understanding of our method.

REFERENCES

[1] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Vianilos, M. Mueller-Freitag et al., "The "something something" video database for learning and evaluating visual common sense.” in ICCV, vol. 1, no. 4, 2017, p. 5.
[2] Y. Li, Y. Li, and N. Vasconcelos, “Resound: Towards action recognition without representation bias,” in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 513–528.
[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in 2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009, pp. 248–255.
[4] C. Feichtenhofer, A. Pinz, and A. Zisserman, “Convolutive two-stream network fusion for video action recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1933–1941.
[5] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould, “Dynamic image networks for action recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3034–3042.
[6] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, “Temporal segment networks: Towards good practices for deep action recognition,” in European conference on computer vision. Springer, 2016, pp. 20–36.
[7] N. Hussein, E. Gavves, and A. W. Smeulders, “Timeception for complex action recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 254–263.
[8] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 4489–4497.
[9] K. Hara, H. Katoaka, and Y. Satoh, “Learning spatio-temporal features with 3d residual networks for action recognition,” in Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017, pp. 3154–3160.
[10] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, “A closer look at spatiotemporal convolutions for action recognition,” in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2018, pp. 6450–6459.
[11] Z. Qiu, T. Yao, and T. Mei, “Learning spatio-temporal representation with pseudo-3d residual networks,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5533–5541.
[12] C. Feichtenhofer, H. Fan, J. Malik, and K. He, “Slowfast networks for video recognition,” in Proceedings of the IEEE conference on computer vision, 2019, pp. 6202–6211.
[13] J. Carreira and A. Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6299–6308.
[14] K. Hara, H. Katoaka, and Y. 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, 2018, pp. 6546–6555.
[15] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in Thirty-first AAAI conference on artificial intelligence, 2017.
[16] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[17] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, “Temporal segment networks for action recognition in videos,” IEEE transactions on pattern analysis and machine intelligence, vol. 41, no. 11, pp. 2740–2755, 2018.
[18] C. Luo and A. L. Yuille, “Grouped spatial-temporal aggregation for efficient action recognition,” in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 5512–5521.
[19] J. Lin, C. Gan, and S. Han, “Tsm: Temporal shift module for efficient video understanding,” in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 7083–7093.
[20] Y. Li, B. Ji, X. Shi, J. Zhang, B. Kang, and L. Wang, “Tea: Temporal excitation and aggregation for action recognition,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 909–918.
[21] B. Jiang, M. Wang, W. Gan, W. Wu, and J. Yan, “Ssm: Spatiotemporal and motion encoding for action recognition,” in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 2000–2009.
[22] X. Wang and A. Gupta, “Videos as space-time region graphs,” in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 399–417.

[23] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in Advances in neural information processing systems, 2014, pp. 568–576.

[24] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” arXiv preprint arXiv:1511.07122, 2015.

[25] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, “Rethinking spatialtemporal feature learning: Speed-accuracy trade-offs in video classification,” in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 305–321.