Spatiotemporal Fusion in 3D CNNs: A Probabilistic View

Yizhou Zhou*[1] Xiaoyan Sun†[2] Chong Luo[2] Zheng-Jun Zha[1] Wenjun Zeng[2]
[1]University of Science and Technology of China [2]Microsoft Research Asia
zyz0205@mail.ustc.edu.cn, zhazj@ustc.edu.cn {xysun,cluo,wezeng}@microsoft.com

Abstract

Despite the success in still image recognition, deep neural networks for spatiotemporal signal tasks (such as human action recognition in videos) still suffers from low efficacy and inefficiency over the past years. Recently, human experts have put more efforts into analyzing the importance of different components in 3D convolutional neural networks (3D CNNs) to design more powerful spatiotemporal learning backbones. Among many others, spatiotemporal fusion is one of the essentials. It controls how spatial and temporal signals are extracted at each layer during inference. Previous attempts usually start by ad-hoc designs that empirically combine certain convolutions and then draw conclusions based on the performance obtained by training the corresponding networks. These methods only support network-level analysis on limited number of fusion strategies. In this paper, we propose to convert the spatiotemporal fusion strategies into a probability space, which allows us to perform network-level evaluations of various fusion strategies without having to train them separately. Besides, we can also obtain fine-grained numerical information such as layer-level preference on spatiotemporal fusion within the probability space. Our approach greatly boosts the efficiency of analyzing spatiotemporal fusion. Based on the probability space, we further generate new fusion strategies which achieve the state-of-the-art performance on four well-known action recognition datasets.

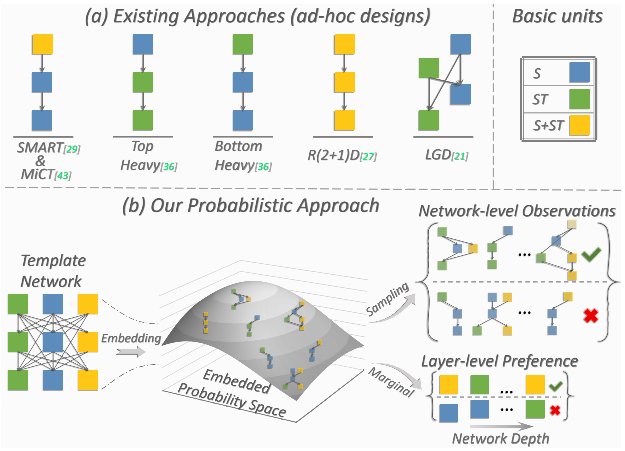


Figure 1: Spatiotemporal fusions in 3D CNNs. (a) Exemplified fusion methods reported in the literature, which are designed empirically and evaluated by training each corresponding network. (b) The proposed probabilistic approach. We propose to analyze the spatiotemporal fusion by finding a probability space where each individual fusion strategy is considered as a random event with a meaningful probability. We first introduce a template network based on basic fusion units to support a variety of fusion strategies. We then embed all possible fusion strategies into the probability space defined by the posteriori distribution over fusion strategy. As a result, various fusion strategies can be evaluated/analyzed without separate network training to obtain network-level observations and layer-level preference. Here S, ST and $S + ST$ are basic fusion units instantiated by 2D, 3D, and a mix of 2D/3D convolutions, respectively.

1. Introduction

For numerous video applications, such as action recognition [31, 43, 33], video annotation [41] and person re-identification [37], spatiotemporal fusion is an integral component. Taking action recognition as an example, the spatiotemporal fusion in deep networks can be roughly classified into two main categories: fusion/ensemble of two modalities (*i.e*, spatial semantics in RGB and temporal dynamics in optical flow) in a two-stream architecture [31, 23] and fusion of spatial and temporal clues in single-stream 3D CNNs [29, 43]. In this paper, we focus on the latter.

Conceptually, 3D CNNs are capable of learning spatiotemporal features responding to both appearance and movement in videos. Recent research also shows that pure 3D CNNs can outperform 2D ones on large scale benchmarks [7]. However, we still observe noticeable variations in accuracy by employing additional spatial or temporal

∗ This work was performed while Yizhou Zhou was an intern with Microsoft Research Asia. † Corresponding author.

feature learning explicitly in 3D CNNs. As shown at the top of Fig. 1, different spatiotemporal fusion strategies [29, 21, 36, 27, 43] have been studied and recommended for action recognition. They explore spatial semantics and temporal dynamics in videos through the combinations of different types of basic convolution unit at each layer in 3D CNNs. Though with different conclusions, these works have one thing in common - they draw conclusions based on the performance of networks employing one or several fusion strategies designed empirically [27, 36, 26]. Each fusion strategy is predefined, fixed, and evaluated in each individual network, leading to a network-level analysis of fusion strategies. Due to the proliferation of combinations and prohibitive computational costs, it is difficult for existing solutions to simulate a great number of fusion strategies for evaluation, nor can they support fine-grained and layer-level analysis.

In this paper, we propose to analyze the spatiotemporal fusion in 3D CNNs from a different point of view, *i.e.*, a probabilistic one. To be specific, we make the spatiotemporal fusion analysis an optimization problem, aiming to find a probability space where each individual fusion strategy is treated as a random event and assigned with a meaningful probability. The probability space will be constructed to meet the following requirements. First, the effectiveness of each spatiotemporal fusion strategy (event) can be *easily* derived from the probability space, so that we can analyze all the fusion strategies based on the derived effectiveness rather than training each network defined by each fusion strategy. Second, from the probability which is closely correlated with the performance of each fusion strategy, it should be able to deduce the layer-level metrics of the fusion efficiencies, making it possible to perform layer-level, fine grained analysis of fusion strategies. Now, the question becomes how we build this probability space.

Recent research shows that optimizing a neural network with dropout (applied on every channel of kernel weights) is mathematically equivalent to the approximation to the posteriori distribution over the network weights [5] and architectures [42]. It inspires us to construct the probability space via dropout in 3D CNNs. In our approach, we propose to first design a template network based on basic fusion units. We define the basic unit as different forms of spatiotemporal convolutions in 3D CNNs, *e.g.*, spatial, spatiotemporal, and spatial+spatiotemporal convolutions, as illustrated in Fig. 1. The probability space can then be defined by the posteriori distribution on different sub-networks (fusion strategies) along with their associated kernel weights in the template network. Note that in our fusion analysis, we need to approximate posteriori distribution on basic fusion units rather than on kernels as in [5]. Therefore, based on the variational Dropout [15] and DropPath [16], we present a Variational DropPath (v-DropPath) by using a variational distribution which factorizes over the probability of the dropout operations that are applied on every basic fusion unit. Then the posterior distribution can be inferred by minimizing the Kullback-Leibler (KL) divergence between the variational distribution and the posteriori distribution, which proves to be equivalent to optimizing the template network with the v-DropPath. We will show that such a probability space fully satisfies the two requirements mentioned above in Section 3.1 and 3.3.

Once we obtain such distribution, we acquire a variety of fusion strategies from the template network by executing v-DropPath *w.r.t.* its optimized drop probability. Those fusion strategies can be directly evaluated without training. In addition, we also utilize the derived probability space to provide numerical measurements for layer-level spatiotemporal fusion preference.

Experimental results show that our proposed probabilistic approach can produce very competitive fusion strategies to obtain state-of-the-art results on four widely used databases on action recognition. It also provides general and practical hints on the spatiotemporal fusion that can be applied to 3D networks with different backbones, such as ResNet[9], MobileNet[22], ResNeXt[35] and DenseNet[10], and achieve good performance.

In summary, our work has four main contributions:

1. We are the first to investigate the spatiotemporal fusion in 3D CNNs from a probabilistic view. Our proposed probabilistic approach enables a highly efficient and effective analysis on varieties of spatiotemporal fusion strategies. The layer-level fine-grained numerical analysis on spatiotemporal fusion also becomes possible.

2. We propose the Variational DropPath to construct the desired probability space in an end-to-end fashion.

3. New spatiotemporal fusion strategies are constructed based on the probability space and achieve the state-of-the-art performance on four well-known action recognition datasets.

4. We also show that the hints on spatiotemporal fusion obtained from the probability space are generic and suitable for benefiting different backbone networks.

2. Related Work

Spatiotemporal fusion has been widely investigated in various tasks and frameworks [21, 18, 44]. In this paper, we choose one of its typical scenarios, *i.e.*, action recognition, to discuss the related work. We further roughly group the spatiotemporal fusion methods for action recognition into two categories: fusion in two-stream (RGB and optical flow) CNNs and fusion in single 3D CNNs. Due to space limitations, here we review only the most related work - spatiotemporal fusion in single 3D CNNs.

There exists a considerable body of literature on spatiotemporal fusion in 3D CNNs. Some of these works show that the efficiency of 3D CNNs can be improved by empirically decoupling the spatiotemporal feature learning in a specific way [29, 3, 21, 43, 4, 45, 2, 13]. For example, Wang et al. [29] present the fusion method that utilizes 3D convolution with square-pooling to capture the appearance-independent relation and 2D convolution to capture the static appearance information. These two features are then concatenated and fed into a 1x1 convolution to form new spatiotemporal features. Results show that this fusion method can significantly improve the performance with model size and FLOPs similar to the original 3D architecture. Feichtenhofer et al. [3] also propose a fusion approach which combines the 3D and 2D CNNs. They use 2D convolution (with more channels) to capture rich spatial semantics from individual frames at lower frame rate, and factorized 3D convolution to extract motion information from frames at high temporal resolution which is fused by lateral connection to the 2D semantics. Zhou et al. [43] present a mixed 3D/2D convolutional tube, MiCT-block, which integrates 2D CNNs with 3D convolution via both concatenated and residual connections in 3D CNNs. It encourages each 3D convolution in 3D network to extract temporal residual information by adding its outputs to the spatial semantic features captured by 2D convolutions.

Instead of presenting one specific fusion strategy, some other work investigates the spatiotemporal fusion in 3D CNNs by evaluating a group of pre-defined fusion methods [27, 36, 26]. For instance, four fusion methods are constructed, trained and evaluated individually in [36] including bottom-heavy-I3D, top-heavy-I3D as shown in Fig.1. More fusions such as mixed convolutions and reversed mixed convolutions are investigated in a similar way in [27, 26]. Although with meaningful observations, these methods can only analyze a limited number of fusion strategies, provide network-level hints, and suffer from huge computational costs.

In contrast to all the above presented methods, in this paper, we propose to construct a probabilistic space that encodes all possible spatiotemporal fusion strategies under a predefined network topology. It not only provides a much more efficient way to analyze a variety of fusion strategies without training them individually, but also facilitates the fine-grained numerical analysis on the spatiotemporal fusion in 3D CNNs.

3. Spatiotemporal Fusion in Probability Space

We observe that a fusion strategy in an L-layer 3D CNN can be expressed with a set of triplets $\{(l, \mathbf{v}, u)\}_L$, where l $(1 \leq l \leq L)$ is the layer index, $\mathbf{v}$ is a binary vector of length $l-1$ denoting the features from which layer/layers will be used, and u $(u \in U)$ denotes the basic fusion units employed in the current layer. Here U is defined by a set of basic fusion units. For example, U can be the combination of three modes, Spatial (S), temporal (T), and spatiotemporal (ST), *i.e.*, $U = \{S, T, ST, S+T, S+ST, T+ST, S+T+ST\}$. As concrete examples, existing fusion strategies can be well represented by the triplets, *e.g.*, top-heavy structure [36], SMART-block[29]/MiCT-block [43] and global diffusion structure [21], as shown in Fig. 2, respectively.

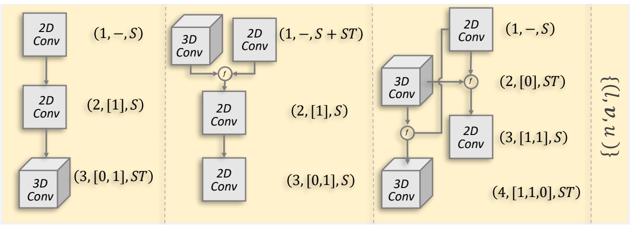


Figure 2: Exemplified triplet representations $\{(l, \mathbf{v}, u)\}$ of three spatiotemporal fusion strategies reported in literature.

3.1. The Probability Space

As discussed in the introduction, we construct the probability space with the posteriori distribution over different fusion strategies along with their associated kernel weights. In the probability space, $\mathcal{M} = \{(l, \mathbf{v}, u)\}_L$ should be a random event. We also define $W_\mathcal{M}$ to be the kernel weight of the corresponding strategy $\mathcal{M}$, which is also a random event in such space. Therefore, we give the full definition of the probability space denoted with $(\Omega, \mathcal{B}, \mathcal{F})$, where

- Sample space $\Omega = \{(\mathcal{M}, W_\mathcal{M})\}$, which is the set of all possible outcomes from the probability space.

- A set of events $\mathcal{B} = \{(\mathcal{M}, W_\mathcal{M})\}$, where each event is equivalent to one outcome in our case.

- Probability measure function $\mathcal{F}$. We use the posteriori distribution to assign probabilities to the events as

$$\mathcal{F} := \mathcal{P}(\mathcal{M}, W_\mathcal{M} \mid \mathbb{D}), \tag{1}$$

  where $\mathbb{D} = \{X, Y\}$ indicates the data samples X and ground-truth label Y used for training.

In this probability space, various fusion strategies and their associated kernel weights are sampled as pairs and we can make direct evaluation without training. The overall performance of one strategy can be obtained only at the cost of network testing. Therefore, the first requirement for the probability space is satisfied. Now, The core of embedding spatiotemporal fusion strategies into such probability space is to derive the measure function defined in Eq. 1.

3.2. Embedding via Variational DropPath

It is hard to obtain the posteriori distribution in Eq. (1), as usual. In our approach, we present a variational Bayesian method to approximate it. We first build a template network based on the basic fusion units that will be studied in the spatiotemporal fusion. For instance, we can design a densely connected 3D CNN with $U = \{S, ST, S+ST\}$, as shown in Fig. 1. We then incorporate a variational distribution that factorizes over every basic unit in the template network, which are re-parameterized with kernel weight multiplying a dropout rate. We further propose the v-DropPath inspired by [15, 5, 42] that enables us to minimize the KL distance between the variational distribution and the posteriori distribution via training the template network. More details will be presented below.

By incorporating the template network, the posterior distribution in Eq. (1) can be converted to

$$\mathcal{P}(\mathcal{M}, W_{\mathcal{M}} \mid \mathbb{D}) \rightarrow \mathcal{P}(\widehat{\mathcal{M}} \circ W_T \mid \mathbb{D}), \tag{2}$$

where $\circ$ is the Hadamard product (with broadcasting), $\widehat{\mathcal{M}} \in (0,1)^{L\times L\times 3}$ is a binary random matrix and $\widehat{\mathcal{M}}(l,i,u) = 1/0$ denotes that the feature from the layer i and the fusion unit u is enabled/disabled at layer l in the template network, respectively. $W_T \in \mathbb{R}^{L\times L\times 3\times V}$ denotes the random weight matrix of the template network, where we use V to denote kernel shape for simplicity. This conversion actually integrates the kernel weights into fusion strategies. Since we can fully recover the $\mathcal{M}$ from the embedded version $\widehat{\mathcal{M}} \circ W_T$ (it is because the kernel is defined in real number field, the probability of being zero for every element can be ignored), the first requirement is still satisfied.

We then approximate the posteriori distribution by minimizing the KL divergence

$$KL(\mathcal{Q}(\widehat{\mathcal{M}} \circ W_T) \mid\mid \mathcal{P}(\widehat{\mathcal{M}} \circ W_T \mid \mathbb{D})), \tag{3}$$

where $\mathcal{Q}(\cdot)$ denotes a variational distribution. Instead of factorizing the variational distribution over convolution channels as in [5], we factorize $\mathcal{Q}(\widehat{\mathcal{M}} \circ W_T)$ over fusion units in each layer as

$$\prod_{l,i,u} q(\widehat{\mathcal{M}}(l,i,u) \cdot W_T(l,i,u,:)). \tag{4}$$

By re-parameterising the $q(\widehat{\mathcal{M}}(l,i,u) \cdot W_T(l,i,s,:))$ with $\epsilon_{l,i,u} \cdot w_{l,i,u}$, where $\epsilon_{l,i,u} \sim Bernoulli(p_{l,i,u})$ and $w_{l,i,u}$ is the deterministic weight matrix associated with the random weight matrix $W_T(l,i,u,:)$, minimizing Eq. 3 is approximately equivalent to minimizing

$$-\frac{1}{N}\log\mathcal{P}(Y \mid X, w\cdot\epsilon) + \frac{1}{N}\sum_{l,i,u} p_{l,i,u}\log p_{l,i,u} + \sum_{l,i,u}\frac{(k_{l,i,u})^2(1-p_{l,i,u})}{2N}\|w_{l,i,u}\|^2, \tag{5}$$

where $k_{l,i,u}$ is a pre-defined length-scale prior and N is the number of training samples. The gradients w.r.t. the Bernoulli parameters p are computed through Gumbel-Softmax [12]. For step-by-step proofs of Eq. 5, please refer to our supplementary material.

Eq. 5 reveals that approximating the posteriori distribution can be achieved by training the template 3D network where each spatial or temporal convolutions is masked by a logit ϵ subject to Bernoulli distribution with probability p. It is exactly the drop-path proposed in [16]. But here both the network weight and the drop rate need to be optimized. We adopt Gumbel-Softmax for the indifferentiable Bernoulli distribution to enable a gradient-based solution. Please find more details in supplementary material.

3.3. Spatiotemporal Fusion

Once the probability space defined by the posteriori distribution is obtained, we can investigate the spatiotemporal fusion very efficiently at both the network and layer levels.

Network-level. Conventionally, the network-level fusion strategies are explored by training and evaluating each individual network defined by one fusion strategy. In our scheme, we successfully eliminate the individual training and evaluation by using the embedded probability space. We study the fusion strategies by directly sampling a group of strategy and kernel weight pairs $\{(\mathcal{M}, W_{\mathcal{M}})^t \mid t = 1, 2, ...\}$ with

$$\mathcal{M}, W_{\mathcal{M}} \sim \mathcal{P}(\widehat{\mathcal{M}} \circ W_T \mid D_{tr}) \approx \mathcal{Q}(\widehat{\mathcal{M}} \circ W_T). \tag{6}$$

It is doable since each $(\mathcal{M}, W_{\mathcal{M}})^t$ can be fully recovered from the embedded version $\widehat{\mathcal{M}} \circ W_T$. The above sample process is equivalent to randomly choosing $\epsilon_{l,i,u}$ based on the Bernoulli distribution with the optimized $p_{l,i,u}$ as defined in Eq. 5, which is further equivalent to randomly dropping some paths in the template network. The effectiveness of each fusion strategy can then be easily derived from the test performance on a validation dataset. Because the sampling and evaluation are light-weight, our approach can greatly expand both the number and form of fusion strategies for analysis.

Layer-level. The network-level analysis shows the overall effectiveness of different spatiotemporal fusion strategies, but rarely reveals the importance of the fusion strategies at each layer. Interestingly, numerical metrics for such fine-grained, layer-level information are also achievable in our approach. Recall that we factorize the variational distribution in Eq. 4 over different fusion strategies using the reparametrisation trick [15]. We thus can deduce the marginal probability of fusion unit at each layer as

$$\mathcal{P}(\widehat{\mathcal{M}}(l,i,u) = 1 \mid \mathbb{D}) = 1 - \sqrt{p_{l,i,u}}. \tag{7}$$

Please refer to supplementary material for detailed derivation. Eq. 7 suggests that the marginal distribution of a spa-

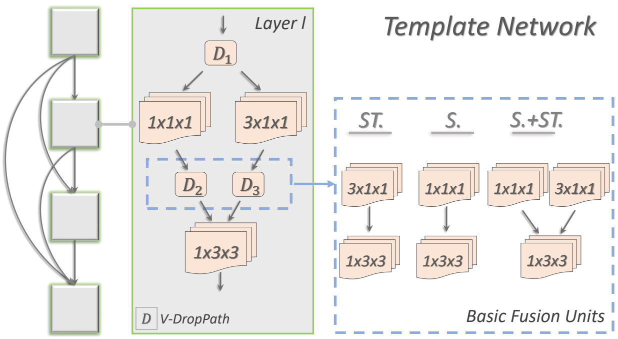


Figure 3: The densely connected template network used in our experiments. In each layer, there are three DropPath (D) operations. The combination of D_2 and D_3 deduces the three basic fusion units $\{S, ST$, and $S + ST$ $\}$. The operations on D_1 and D_2/D_3 correspond to the index i and u in $\epsilon_{l,i,u}$, respectively.

tiotemporal fusion strategy can be retrieved from the optimized dropout probability. It indicates the probability of using a fusion unit among all the possible networks that can interpret the given dataset well and satisfy prior constrains (sparsity in our case). We propose using this number as the indicator of the layer-level spatiotemporal preference. Therefore, the second requirement on the probability space is met, too.

4. Experiments

In this section, we will verify the effectiveness of our probabilistic approach from three aspects. Four action recognition databases are used in the experiments. After the description of experimental setups, we will first show the performance of the fusion strategies obtained by our approach in comparison with those of state-of-the-arts. Then several main observations are provided based on the analysis of different fusion strategies generated from our probability space. At last, we verify the robustness of the obtained spatiotemporal fusion strategies on different backbone networks.

4.1. Experimental Setups

Template Network. Fig. 3 sketches the basic structure of the template network designed for our approach. The template network is a densely connect one that comprises of mixed 2D and 3D convolutions. Here we choose $U = \{S, ST, S + ST\}$ so that the fusion units explored in our approach are conceptually included in most of other fusion methods for fair comparison. We also factorize each 3D convolution with a 1D convolution and a 2D convolution, and use element-wise summation to fuse the 2D and 3D convolutions for simplicity. Besides, we add several transition blocks to reduce the dimension of features and the total number of layers is set to be 121 as in [10]. We put more details of the template network in the supplementary material. In practice, we share the variational probability on the variables i defined in Section. 3 for computational efficiency.

Datasets. We apply the proposed scheme on four well-known action recognition datasets, *i.e.*, Something-Something(V1&V2)[6], Kinetics400[14] and UCF101[24]. Something V1/V2 consist of around 86k/169k videos for training and 12k/25k videos for validation, respectively. Video clips in these two datasets are first-person videos with 174 categories that focus more on temporal modelling. Kinetics400 is a large-scale action recognition database which provides around 240k training samples and 20k validation samples from 400 classes. UCF101 has around 9k and 3.7k videos for training and validation. They are categorized into 101 classes. Both the Kinetics400 and the UCF101 contain complex scene and object content in video clips and have large temporal redundancy.

Training. As mentioned before, we approximate the posteriori distribution of different fusion strategies by training the template network with v-DropPath. We initialize the drop rate of each convolution operation as 0.1. We train the template network with 90 epochs for Something-Something(V1&V2)/UCF101 and 110 epochs for Kinetics400, respectively. The batch size is 64 for Kinetics and 32 for the others. The initial learning rates are 0.005 (Something&UCF) and 0.01 (Kinetics) and we decay them by multiplying 0.1 at 40th, 60th, 80th epochs for Something/UCF and 40th, 80th epochs for Kinetics. The video frames are all resized to 256 (short edge) and randomly cropped to 224x224. The length-scale prior k in Eq. 5 is determined by grid search, where $k = 250$ for SomethingV1, $k = 10$ for Kinetics400 and $k = 50$ for the rest. In practice, warmup is used before training the template network with v-DropPath, *i.e.*, removing all the v-DropPath operations and training the template network from scratch for 50 epochs. All experiments are conducted with distributed settings and synchronized Batch Normalization [11] on multiple (8-32) v100 GPUs with 32G memory.

Sampling and Inference. We derive various spatiotemporal fusion strategies from the probability space through sampling different combinations of spatiotemporal convolutions w.r.t. the drop probability of v-DropPath. The sampled strategies are directly evaluated on validation dataset.

During the inference of each spatiotemporal fusion strategy, we resize the short edge of the input video frames to 256 and make center crop to get a 256x256 region. We uniformly sample multiple clips in a video and average the prediction scores to obtain video level predictions. The number of clips varies from dataset to dataset and will be discussed along with the results.

Table 1: Performance evaluation on Something-Something V1. Im./K.400 denote ImageNet/Kinetics400 pre-training.

Method	Backbone	Extra Mod.	Pretrain	#F	FLOPs	#Param.	Top-1	Top-5
TSN[31]	BNInception	-	Im.	8	16G	10.7M	19.5%	-
TSN[17]	ResNet50	-	Im.	8	33G	24.3M	19.7%	46.6%
TRN-Multiscale[40]	BNInception	-	Im.	8	16G	18.3M	34.4%	-
TRN-Multiscale[17]	ResNet50	-	Im.	8	33G	31.8M	38.9%	68.1%
Two-stream TRN[40]	BNInception	-	Im.	16	-	36.6M	42.0%	-
TSM[17]	ResNet50	-	Im.	16	65G	24.3M	47.2%	77.1%
TrajectoryNet[39]	3D Res.18	Y	Im.+K.400	-	-	x	47.8%	-
STM[13]	3D Res.50	Y	Im.	16	66.5G	24.0M	49.8%	-
Non-local I3D[32]	3D Res.50	Y	Im.	64	336G	35.3M	44.4%	76.0%
Non-local I3D + GCN[32]	3D Res.50+GCN	Y	Im.	64	606G	62.2M	46.1%	76.8%
S3D-G[36]	3D BNincept.+gate	Y	Im.	64	71G	11.6M	48.2%	78.7%
I3D[32]	3D Res.50	N	Im.	64	306G	28.0M	41.6%	72.2%
I3D[36]	3D BNIncept.	N	Im.	64	108G	12.0M	45.8%	76.5%
S3D[36]	3D BNIncept.	N	Im.	64	66G	8.77M	47.3%	78.1%
ECO[45]	BNIncept.+3DRes.18	N	Im.+K.400	8	32G	47.5M	39.6%	-
ECO[45]	BNIncept.+3DRes.18	N	Im.+K.400	16	64G	47.5M	41.4%	-
ECO Lite[45]	BNIncept.+3DRes.18	N	Im.+K.400	92	267G	150M	46.4%	-
Ours	3D DenseNet121	N	Im.	16	31G	21.4M	**50.2%**	**78.9%**

Table 2: Ablation studies on the selected spatiotemporal fusion strategies from our probability space.

Strategy Dataset	S	ST	S+ST	Opt
SomethingV1	41.8%	47.5%	46.5%	**50.2%**
SomethingV2	55.1%	60.5%	59.5%	**62.4%**
UCF101	83.6%	83.1%	84.2%	**84.2%**
Kinetics400	67.8%	68.3%	69.7%	**71.7%**

4.2. Ablation Study

In order to demonstrate the effectiveness of our probability space, for each dataset, we sample 100 fusion strategies from the constructed space and choose the best one according to the performance on the held-out validation dataset. We denote the best strategy as 'Optimized'(Opt). We then compare it with its counter-part strategies 'S','ST', and 'S+ST' in Fig. 2, which are designed with one fixed corresponding basic fusion unit, S, ST, or $S + ST$, at all layers, respectively. It can be observed that our probability space can generate better strategies on all the dataset. Our 'Opt' method even outperforms its counter-part 'ST+S' which has more parameters and higher FLOPs.

4.3. Comparisons with the State-of-the-arts

Our proposed method analyzes the spatiotemporal fusion strategies from the perspective of the probability. It not only enables an advance analysis approach, but also achieves high-performance spatiotemporal fusion strategies. In this section, we compare the strategies drawn from the probability space with state-of-the-art fusion methods on four action recognition datasets. Our approach has very competitive performance, *i.e.*, performing the best among all the schemes on three of these datasets and obtaining the second best on UCF101, even though some of the compared results are achieved with better backbones and/or with extra modules such as non-local, motion encoder, or gated functions.

Something-Something V1&V2. Table. 1 exhibits the performance of different spatiotemporal fusion methods on Something V1 dataset. It shows that our approach leads to the fusion strategy that outperforms all the other schemes including so far the most advanced 3D network S3D by a large margin with 50% fewer FLOPs and frames. Surprisingly, it performs even better than those methods with carefully designed functional modules, *e.g.*, STM employs a channel-wise motion module to explicitly encode motion information, and Non-local I3D + GCN explicitly incorporates the object semantics with graphs. Similar results can be observed on the recently released dataset Something V2. As shown in Table. 3, our fusion strategies significantly outperform the conventional I3D solutions and its bottom-heavy and top-heavy counterparts which incorporates 3D convolutions in bottom layers and top layers, respectively. We employ ImageNet pre-training for both datasets and our fusion strategy can achieve higher accuracy than those pre-trained on the large-scale dataset Kinetics such as ECO.

Kinetics400. Accuracy achieved by different fusion methods on Kinetics400 are reported in Table 4. In order to make apple-to-apple comparisons, all methods are trained from scratch. It can be observed that our configuration of spatiotemporal fusion outperforms the second best R(2+1)D on Top1 accuracy with 97% fewer FLOPs , where R(2+1)D is a 3D network that uses ResNet34 as backbone. Compared with R(2+1)D, we actually utilize more spatial convolutions in the shallow layers as can be viewed in Fig. 4.

Table 3: Performance comparison with state-of-the-art results on Something-Something V2.

Method	Val. Top-1	Val. Top-5
TSN[17]	30.0%	60.5%
MultiScale TRN[40]	48.8%	77.6%
Two-stream TRN[40]	55.5%	83.1%
TSM(ImageNet+ Kinetics400)[17]	59.1%	85.6%
TSM dual attention[34]	55.5%	82.0%
I3D-ResNet50[34]	43.8%	73.2%
2D-3D-CNN w/ LSTM [19]	51.6%	-
Ours (ImageNet)	**62.9**%	**88.0**%

Table 4: Performance comparison with the state-of-the-art results of different spatiotemporal fusions in 3D architectures on Kinetics400 trained from the scratch.

Method	Backbone	FLOPs	Top1	Top5
STC[2]	R.Xt101	N/A × N/A	68.7%	88.5%
ARTNet[29]	ResNet18	23.5G × 250	69.2%	88.3%
R(2+1)D[27]	ResNet34	152G × 115	72.0%	90.0%
S3D*[36]	BNIncept.	66.4G × 250	69.4%	89.1%
I3D[1]	BNIncept.	216G × N/A	68.4%	88.0%
ECO[45]	custom	N/A × N/A	70.0%	89.4%
3DR.Xt[7]	R.Xt101	N/A × N/A	65.1%	85.7%
Disentan.[38]	BNIncept.	N/A × N/A	71.5%	89.9%
StNet [8]	ResNet101	311G x 1	71.4%	-
Ours	Dense.121	254G × 2	**72.5**%	**90.3**%

UCF101. Since UCF101 has only 9k training videos, we make evaluations with the ImageNet pre-training and Kinetics400 pre-training, respectively. When incorporating ImageNet pre-training only, our fusion strategy produces the most advanced results, which has 1.5% higher accuracy than the I3D that performs pure spatiotemporal fusions. When using Kinetics400 as pre-training dataset, the overall performance is still state-of-the-art. Please note that we do not employ any extra functional module here, so the performance is slightly worse (0.3%) than the most advanced 3D networks S3D-G that incorporates attention mechanism.

4.4. Observations

We visualize the strategies derived from the probability space that have the highest accuracy on the test datasets in Fig. 4. We also illustrate the marginal probability of using different basic units in each layer based on Eq. 7. The amplitude of bars in blue, green and yellow indicates the marginal probability of using the units S, ST and $S + ST$ in each layer, respectively. The dotted-line in orange shows the selected layer-level basic fusion units that produce the best accuracy. From these figures, we observe that

Observation I. As indicated by the colored bars, the unit $S + ST$ has higher marginal probability in the lower-level feature learning compared with the other two units. The dotted line in orange also shows a similar trend. The $S + ST$ unit has the highest percentage of total usage in all the fusion units, especially in the lower layers. It suggests that a proper spatiotemporal fusion strategy can be designed based on $S + ST$ units, particularly in lower layers.

Table 5: Performance comparison with the state-of-the-art results on UCF101. Im., S.1M and K.400 denote ImageNet, Sport1M and Kinetics400, respectively. Our methods with ResNeXt50 and Inception backbones are designed according to the hints we observe from the probability space. Please refer to Section 4.4 and 4.5 for details.

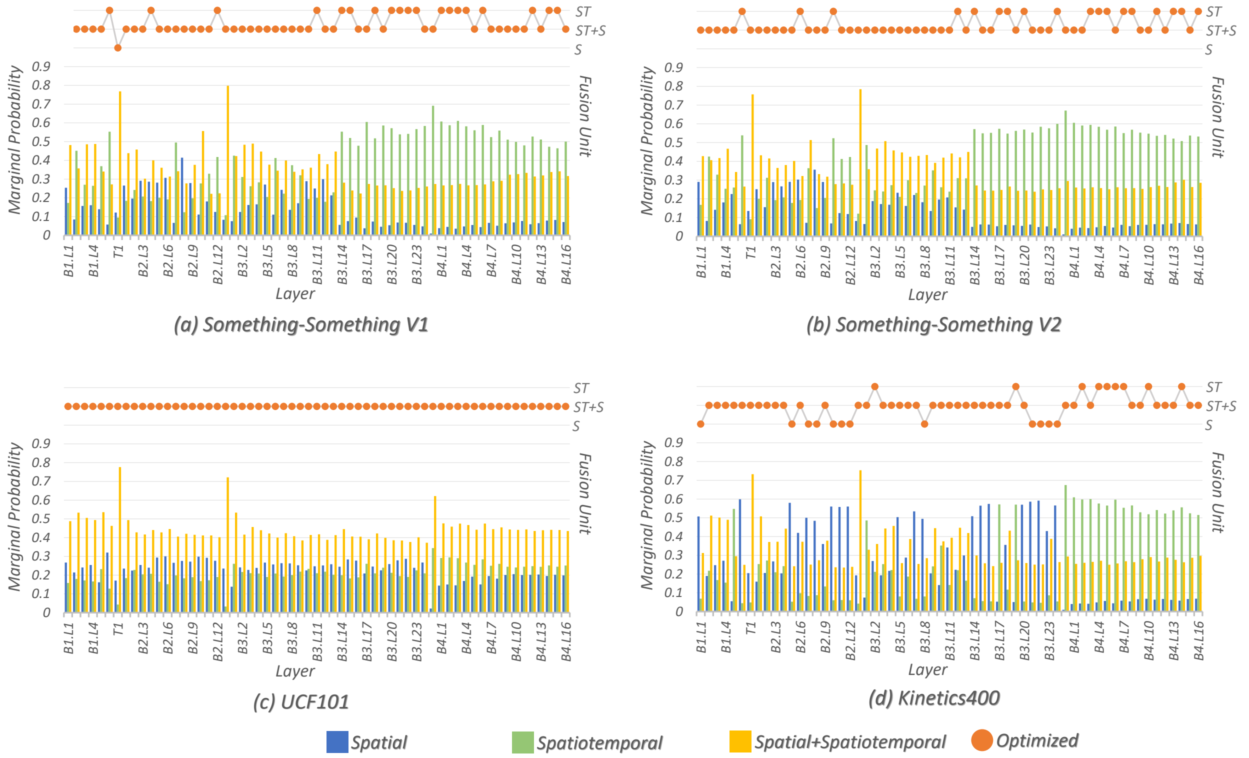
Method	Pre.	Backbone	Top-1
TDD[30]	Im.	VGG-M	82.8%
C3D[25]	Im.	3DVGG11	44.0%
LTC[28]	Im.	3DVGG11	59.9%
ST-ResNet[4]	Im.	3DRes.50	82.3%
I3D[1]	Im.	3DIncept.	84.5%
Ours	Im.	3DDenseNet121	**85.0%**
Ours	Im.	3DRexNeXt50	**86.0%**
Res3D[26]	S.1M	3DRes.18	85.8%
P3D[20]	S.1M	3DRes.199	88.6%
MiCT[43]	S.1M	3DIncept.	**88.9%**
Res3D[26]	K.400	3DRes.18	89.8%
TSN[31]	K.400	Incept.V3	93.2%
I3D[1]	K.400	3DIncept.	95.6%
ARTNet[29]	K.400	3DRes.18	94.3%
R(2+1)D[27]	K.400	3DRes.34	**96.8%**
S3D-G[36]	K.400	3DIncept.	**96.8%**
3DResNeXt101[7]	K.400	-	94.5%
STM[13]	K.400	3D Res.50	96.2%
STC[2]	K.400	3DResNext101	96.5%
Ours	K.400	3DDenseNet121	94.5%
Ours	K.400	3DIncept.	96.5%

Observation II. More ST units are preferred in higher layers as there is a higher marginal probability on the ST unit in the higher-level feature learning (except on UCF101 which will be discussed below).

Observation III. Additional S units could be beneficial when scene semantics are complex. For instance, Kinetics400/UCF101 contain videos in the wild with 400/101 different categories, respectively. The scene content is more complex than that in the first-person videos in Something-Something. By comparing Fig. 4 (c) and (d) with the others, it shows that more S or $S + ST$ units are selected.

4.5. Generalization

We further discuss the generalization of our observations as well as the selected fusion strategies. We extend our fusion strategies to three backbone networks including ResNet50[9], and ResNeXt50/ResNeXt101[35]. They differ from each other in terms of topology, parameter size and FLOPs. We report clip-level accuracy on Something V1 for quick comparison. Please find more results and discussions on other backbone networks in the supplementary material.



(a) Something-Something V1

(b) Something-Something V2

(c) UCF101

(d) Kinetics400

Figure 4: Visualization of our spatiotemporal fusion strategies and marginal probabilities of layer level fusion units. On the top of each sub-figure, we show the fusion strategy derived from the probability space that has the highest accuracy by the dotted line in orange. Three units, S, ST and $S+ST$, are involved as shown on the right side of each sub-figure. The amplitude of bars in blue, green and yellow indicates the marginal probability of using the basic units S, ST and $S+ST$ in each layer, respectively. The x-axis indexes layers, where B denotes dense blocks and L is the layer index in the block.

Table 6: Generalization of the observations. The fusion strategies 'Opt' for each backbone are straightforwardly designed based on the observations.

Strategy / Net.	S	ST	S+ST	Opt
3D ResNet50	33.8%	40.1%	38.9%	**41.2%**
3D ResNeXt50	35.2%	42.1%	40.7%	**43.6%**
3D ResNeXt101	36.6%	42.7%	42.3%	**44.0%**

We employ four different fusion strategies 'Opt', 'S+ST', 'S' and 'ST' as defined in Section 4.2 for comparison. Note that here the fusion strategy denoted by 'Opt' is not optimized using our probabilities approach but straightforwardly designed based on our observations. Specifically, we construct the fusion strategy 'Opt' according to Fig. 4 (a) and (b), which uses $S+ST$ unit in both the first half and the last three layers, and ST unit in the remaining layers. As shown in Table. 6, the fusion method 'Opt' performs the best among all the evaluated fusion strategies.

5. Conclusion and Discussion

In this paper, we convert the problem of analyzing spatiotemporal fusion in 3D CNNs into an optimization problem which aims to embed all possible fusion strategies into the probability space defined by the posteriori distribution on each fusion strategy along with its associated kernel weights. Such probability space enables us to investigate spatiotemporal fusion from a probabilistic view, where various fusion strategies are evaluated and analyzed without the needs of individual network training. The numerical measurements on layer-level fusion preference are available. By further proposing the Variational DropPath, the optimization problem can be efficiently solved via training a template network. Experimental results on four action recognition databases demonstrate the effectiveness of our approach. We also observe several useful hints with our probabilistic approach which can be extended to design high performance fusion strategies on different backbones.

Acknowledgement

This work was supported by the National Key R&D Program of China under Grant 2017YFB1300201, the National Natural Science Foundation of China (NSFC) under Grants 61622211, U19B2038 and 61620106009 as well as the Fundamental Research Funds for the Central Universities under Grant WK2100100030.

References

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

[2] Ali Diba, Mohsen Fayyaz, Vivek Sharma, M Mahdi Arzani, Rahman Yousefzadeh, Juergen Gall, and Luc Van Gool. Spatio-temporal channel correlation networks for action classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 284–299, 2018.

[3] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. *arXiv preprint arXiv:1812.03982*, 2018.

[4] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Spatiotemporal residual networks for video action recognition. In *Advances in neural information processing systems*, pages 3468–3476, 2016.

[5] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[6] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *ICCV*, volume 1, page 5, 2017.

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

[8] Dongliang He, Zhichao Zhou, Chuang Gan, Fu Li, Xiao Liu, Yandong Li, Limin Wang, and Shilei Wen. Stnet: Local and global spatial-temporal modeling for action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8401–8408, 2019.

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

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

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[12] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

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

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

[15] Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pages 2575–2583, 2015.

[16] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *arXiv preprint arXiv:1605.07648*, 2016.

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

[18] Jiawei Liu, Zheng-Jun Zha, Xuejin Chen, Zilei Wang, and Yongdong Zhang. Dense 3d-convolutional neural network for person re-identification in videos. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 15(1s):1–19, 2019.

[19] Farzaneh Mahdisoltani, Guillaume Berger, Waseem Gharbieh, David Fleet, and Roland Memisevic. Fine-grained video classification and captioning. *arXiv preprint arXiv:1804.09235*, 2(6), 2018.

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

[21] Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, Xinmei Tian, and Tao Mei. Learning spatio-temporal representation with local and global diffusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12056–12065, 2019.

[22] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[23] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[24] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

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

[26] Du Tran, Jamie Ray, Zheng Shou, Shih-Fu Chang, and Manohar Paluri. Convnet architecture search for spatiotemporal feature learning. *arXiv preprint arXiv:1708.05038*, 2017.

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

[28] Gül Varol, Ivan Laptev, and Cordelia Schmid. Long-term temporal convolutions for action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1510–1517, 2017.

[29] Limin Wang, Wei Li, Wen Li, and Luc Van Gool. Appearance-and-relation networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1430–1439, 2018.

[30] Limin Wang, Yu Qiao, and Xiaoou Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4305–4314, 2015.

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

[32] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 399–417, 2018.

[33] Haoze Wu, Jiawei Liu, Zheng-Jun Zha, Zhenzhong Chen, and Xiaoyan Sun. Mutually reinforced spatio-temporal convolutional tube for human action recognition. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 968–974, 2019.

[34] Tete Xiao, Quanfu Fan, Dan Gutfreund, Mathew Monfort, Aude Oliva, and Bolei Zhou. Reasoning about human-object interactions through dual attention networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3919–3928, 2019.

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

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

[37] Wei Zhang, Shengnan Hu, Kan Liu, and Zhengjun Zha. Learning compact appearance representation for video-based person re-identification. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(8):2442–2452, 2018.

[38] Yue Zhao, Yuanjun Xiong, and Dahua Lin. Recognize actions by disentangling components of dynamics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6566–6575, 2018.

[39] Yue Zhao, Yuanjun Xiong, and Dahua Lin. Trajectory convolution for action recognition. In *Advances in Neural Information Processing Systems*, pages 2204–2215, 2018.

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

[41] Yizhou Zhou, Xiaoyan Sun, Dong Liu, Zhengjun Zha, and Wenjun Zeng. Adaptive pooling in multi-instance learning for web video annotation. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 318–327, 2017.

[42] Yizhou Zhou, Xiaoyan Sun, Chong Luo, Zheng-Jun Zha, and Wenjun Zeng. One-shot neural architecture search through a posteriori distribution guided sampling. *arXiv preprint arXiv:1906.09557*, 2019.

[43] Yizhou Zhou, Xiaoyan Sun, Zheng-Jun Zha, and Wenjun Zeng. Mict: Mixed 3d/2d convolutional tube for human action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 449–458, 2018.

[44] Yizhou Zhou, Xiaoyan Sun, Zheng-Jun Zha, and Wenjun Zeng. Context-reinforced semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4046–4055, 2019.

[45] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. Eco: Efficient convolutional network for online video understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 695–712, 2018.