NUTA: Non-uniform Temporal Aggregation for Action Recognition

Xinyu Li* Chunhui Liu* Bing Shuai Yi Zhu Hao Chen Joseph Tighe
{xxnl, chunhuil,bshuai,yzaws,hxen,jtighe}@amazon.com
Amazon Web Services

Abstract

In the world of action recognition research, one primary focus has been on ***how to*** *construct and train networks to model the spatial-temporal volume of an input video. These methods typically uniformly sample a segment of an input clip (along the temporal dimension). However, not all parts of a video are equally important to determine the action in the clip. In this work, we focus instead on learning* ***where to*** *extract features, so as to focus on the most informative parts of the video. We propose a method called the non-uniform temporal aggregation (NUTA), which aggregates features only from informative temporal segments. We also introduce a synchronization method that allows our NUTA features to be temporally aligned with traditional uniformly sampled video features, so that both local and clip-level features can be combined. Our model has achieved state-of-the-art performance on four widely used large-scale action-recognition datasets (Kinetics400, Kinetics700, Something-something V2 and Charades). In addition, we have created a visualization to illustrate how the proposed NUTA method selects only the most relevant parts of a video clip.*

1. Introduction

A key challenge in action recognition is how to learn a feature that captures the relevant spatial and motion queues in an efficient and compact representation. This problem has been well studied, primarily using convolution neural networks (CNNs) [12], from frame-based methods [16] to segment based temporal information aggregation [35] to the I3D based methods [2]. These methods have primarily focused on **how to** perform feature extraction that captures a complete spatial-temporal description of the video. The majority of these methods treat each frame, or point in time, with equal weight, but not all parts of the video are equally important and thus it is also key that we develop feature extraction methods that can determine **where to** extract features from. Some recent works have started to look at this problem with non-local modules [36], directional convolutions [18] or ensembles of features at different sample rates [8, 42], but these methods still focus more on how to comprehensively extract features rather than where to extract the features. In this work we leverage all the advancements that have been made in the past years on how to extract motion features and focus our attention on the question of **where to** extract these features from.

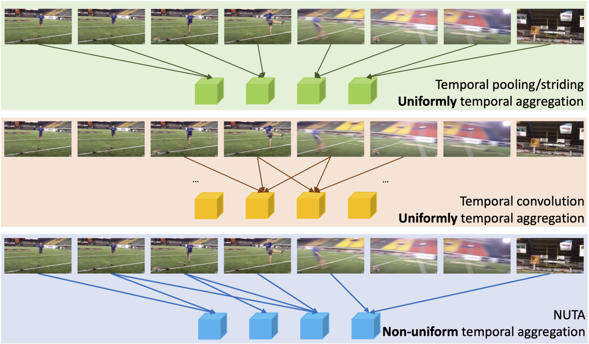
Temporal pooling/striding
Uniformly temporal aggregation
...
...
Temporal convolution
Uniformly temporal aggregation
NUTA
Non-uniform temporal aggregation

Figure 1: A comparison of different temporal modeling methods. The temporal pooling/striding and temporal convolution aggregate features uniformly along time. Our non-uniform temporal aggregation (NUTA) extracts features non-uniformly from informative parts of the video clip.

There are two major challenges to answer the "**where to**" question. First, non-uniformly extracting features efficiently from only informative temporal episodes is challenging as it is required to look at the whole video to determine which parts are informative.

Some recent work [17, 40] have proved that a recognition system can benefit from selecting the informative frames rather than simply taking the uniformly sampled frames as inputs. However, these systems treat the frame selection and feature extraction as two separate stages and thus the frame selection can not benefit from the later feature extraction thus reducing the descriptive power of the network and adding redundancy in the two stages.

[*] Equally contributed.

Second, exchanging information between different feature sampling strategies efficiently and comprehensively remains a challenge. Other works have tackled this using kernels [20, 24, 36], LSTMs [6] or most recently feature pyramids [42] but each of these methods only partially deals with this information exchange problem.

In order to address the first challenge (how to extract feature from only representative temporal instances), we introduce a non-uniform temporal aggregation (NUTA) method. In contrast to the average pooling operator or temporal stride convolution kernels (Figure 1), the NUTA module generates a temporal map that non-uniformly projects T temporal instances to T' ($T' < T$). The NUTA has receptive field in temporal dimension as large as the entire clip so the NUTA module is able to generate clip-level features. Differing from previous two-stage systems that perform frame selection first and then do recognition on the selected frames [17, 40], our NUTA module performs temporal information selection at the feature level in an end-to-end manner. Differing from previous work that focus on "stacking" features of the same kind [8, 36] to construct the comprehensive feature representation, NUTA works by focusing on a sub-set of the frames that are the most informative for the task. To address the second challenge (information exchange between network branches), we propose a two-branch network with a novel temporal synchronization strategy. The two-branch design maintains the uniformly aggregated local feature branch and the non-uniformly aggregated clip-level feature branch separately. Note that the purpose of our two-branch design is to separate the feature descriptors instead of using multiple input modalities to encourage different feature extraction behaviors as in [8, 42]. The temporal synchronization strategy temporally aligns the features from the two branches to allow the information from each branch to flow to the other. Such design not only improves the action recognition performance by removing noise from both the uniform and non-uniform branches, but also reduces the computational requirements of our network, allowing our two-branch design to maintain similar computational cost to the original single branch network.

We tested our model on four of the most popular action recognition datasets: Kinetics 400, Kinetics 700, Something Something V2 and Charades. Our NUTA network achieves the state-of-the-art (SOTA) or comparable performance on all four datasets with less computation cost. We perform a detailed ablation study that shows our non-uniform aggregation method is effective across a wide range of configurations. We also visualized our model to better demonstrate how our NUTA behaves and how it helps answer the "where to" question. To summarize, our contributions are:

1. We propose a NUTA module to answer the "where to" question, which is able to identify and sample only the informative parts of the video clips;
2. We propose a two-branch NUTA network for action recognition, which learns local and clip-level features by fusing the feature extracted from the best in bread video modeling methods and our novel NUTA module;
3. We propose information fusion and temporal synchronization methods that allow both branches to specialize on their own aspects of video modelling while sharing their feaures after each stage of the network;
4. We present experimental results on four common, large scale action recognition datasets, which shows state-of-the-art or comparable performance on each.

2. Related Work

The early video action recognition work used 2D CNNs [16, 29] to extract and aggregate frame-wise features with no effective temporal modeling. Therefore, the question of how to perform temporal modeling became a key area of interest. RNNs and LSTMs [14] were used to model temporal associations among features extracted by 2D networks [19, 43, 5, 21, 15, 23]. However, LSTM based methods often significantly increased the computation with little benefit to accuracy. Segment level predictions methods such as TSN [35] and rank pooling [10, 1] based methods were proposed to better capture the temporal feature evolution at the video level. Although quite successful, these methods still followed the idea of aggregating 2D features instead of performing pixel-level 3D feature learning. More recently, 3D CNNs [2, 32, 34, 41, 33, 27, 25, 34] have gained significant popularity due to their ability to perform spatio-temporal feature modeling. The 3D CNNs effectively address the "how to" question and are now used as basic building blocks for most temporal modeling tasks. Most of recent works have continued to focus on "how to" perform temporal modeling more comprehensively: the non-local network [36] leverages non-local connections to establish pixel level long-term spatio-temporal association; the TEA [44] and TAM [24] methods use a local and global kernel for long-shot term feature learning; the SlowFast [8] and TPN [42] methods ensemble feature at different frame-rate for better performance. There have been a handful of papers that have tried to answer the "where to" question. Early research tried to generate long-term feature with LSTM and temporal attention [31, 6] but none of these methods ended up achieving competitive performance. Two recent methods, SCSampler [17] and multi-agent sampler [40] select only informative clips from a video for action recognition and have been able to demonstrate better performance than uniformly sampling the frames from a video.

Different from these previous work, we answer the "where to" question by proposing the non-uniform temporal aggregation module which learn the feature non-uniformly from informative temporal episodes. Instead of adding features, e.g. from non-local connection or from multiple branches that takes different types of input [36, 8], our network reduce redundant information for better recognition. Different from two-stage system that first select representative frames and run classifier on selected frames [17], our proposed method runs end-to-end and performs feature level information selection.

Many two-branch architectures have been proposed to fuse information from different sources, e.g. input frames at different resolution [16], RGB and optical flow images [35, 29, 9], and video clips at different sample rate [8, 44]. Differing from most previous two-branch systems that fuse features from different input modalities or representations, our two-branch design models both different and complementary features from the same input.

3. Methodology

3.1. Overview

We propose the Non-uniform Temporal Aggregation (NUTA) network that follows a two-branch design with one uniform branch that maintains the relative temporal relations of the input clip and captures temporal local information, and one non-uniform branch for clip-level temporal feature selection (Figure 2). We adopt a 3D convolutional style network for our uniform branch to learn local spatial-temporal features. We propose a novel NUTA module and build our non-uniform branch by stringing multiple NUTA modules together. Each NUTA module is able to model the full clip-level temporal feature by sampling non-uniformly along the temporal dimension. Our NUTA module does not perform any spatial modeling, thus to enable the non-uniform branch to model the clip at different spatial scales. We transfer information between the two branches after each stage. This information exchange between our uniform and non-uniform branches is non trivial because each operates on different samples from the input clip. To overcome this issue we introduce our temporal synchronization strategy, which synchronizes the features from the uniform and non-uniform branches bidirectionally. The rest of this section introduces the components of our NUTA module including the non-uniform down-sampling, feature synchronization and feature fusion strategies.

3.2. Non-uniform Temporal Down-sampling

A core feature of our NUTA network is our non-uniform sampling of the video clip feature along the temporal dimension. This component is to only sample the most informative parts to feature across time. We achieve this via our down-sampling module that is parameterized as a learned projection map, which maps a feature of temporal dimension T to T' ($T' = \frac{T}{2}$ in this paper). Given a feature tensor $\mathbf{F} \in \mathbb{R}^{C\times T\times W\times H}$, we apply a self-attention-like module to learn the projection map $\mathbf{M}$ as:

$$\mathbf{M} = \mathrm{softmax}\left(\Gamma[\phi(\mathrm{pool}(\mathbf{F}))] \times \Gamma[\theta(\mathbf{F})]^T\right) \qquad (1)$$

where $\mathrm{pool}(*)$ denotes temporal max-pooling with kernel size of 2, functions $\phi(*)$ and $\theta(*)$ are convolution operators, and $\Gamma[*]$ denotes a reshape and permutation operation. Then, the clip-level, non-uniformly sampled feature $\mathbf{F}_{\mathbf{NUTA}}$ is generated by applying the temporal projection map as:

$$\mathbf{F}_{\mathbf{NUTA}} = \mathrm{Conv}\left(\Gamma^{-1}[\mathbf{M} \times \Gamma[\delta(\mathbf{F})]]\right) \qquad (2)$$

where the $\delta(*)$ denotes a convolution, the Γ^{-1} is the inverse reshape of Γ, and Conv denotes a convolution to change the channel dimension for information fusion with the next stage. In practice, we use a $(3, 1, 1)$ convolution kernel for $\phi(*)$, $\theta(*)$ and $\delta(*)$. We find such convolution gives a better performance than commonly used $(1, 1, 1)$ kernel, which may due to it gives larger capacity on temporal modeling.

Depending on the complexity of the scene and action classification needed, there may be some set of frames that are most informative for one type of action and a completely different set for another type of action. To allow our network to have the flexibility of picking different sets of temporal periods for each channel we extend our projection map by making it multi-head. Each head is then free to select its own set of frames and extract features from those frames. We implement this using our $\Gamma[*]$ operation that split $\phi(\mathbf{F}) \in \mathbb{R}^{T\times C\times W\times H}$ to n heads along the channels, such that $\Gamma[\phi(\mathbf{F})] \in \mathbb{R}^{n\times T\times \frac{C}{n}WH}$. To make our design computationally efficient, we use the group convolution and empirically set groups as 64 (see section ablation study for details).

Although it may appear similar, our proposed temporal modeling is significantly different from non-local network [36] as our generated projection map $\mathbf{M} \in \mathbb{R}^{n\times\frac{T}{2}\times T}$ represents a temporal association based on spatial features while the non-local kernel $\mathbf{M}' \in \mathbb{R}^{TWH\times TWH}$ represents a pixel level spatial temporal association learned from channels.

3.3. Temporal Synchronization

The output feature from our NUTA module $\mathbf{F}_{\mathbf{NUTA}n} \in \mathbb{R}^{C\times\frac{T}{2}\times W\times H}$ is not compatible with the uniform branch feature ($\mathbf{F}_{\mathbf{res}n} \in \mathbb{R}^{C\times T\times W\times H}$) for two reasons: 1) the temporal dimensions do not match ($\frac{T}{2} \neq T$) and 2) they are not sampled from the same set of frames. To handle this incompatibility we propose to synchronize the features from the non-uniform branch to the uniform branch (Figure 2 yellow arrow) as:

$$\mathbf{F}_{\mathbf{res}} = \mathrm{Conv}\left(\Gamma^{-1}[\mathbf{M} \times \Gamma[\zeta(\mathbf{F}_{\mathbf{res}})]]\right) + \mathrm{pool}(\mathbf{F}_{\mathbf{res}}) \qquad (3)$$

where ζ denotes a 3D convolution with kernel $(3, 1, 1)$ similar to θ and ϕ. Operation $\mathrm{Conv}\left(\Gamma^{-1}[\mathbf{M} \times \Gamma[\zeta(\mathbf{F}_{\mathbf{res}})]]\right)$

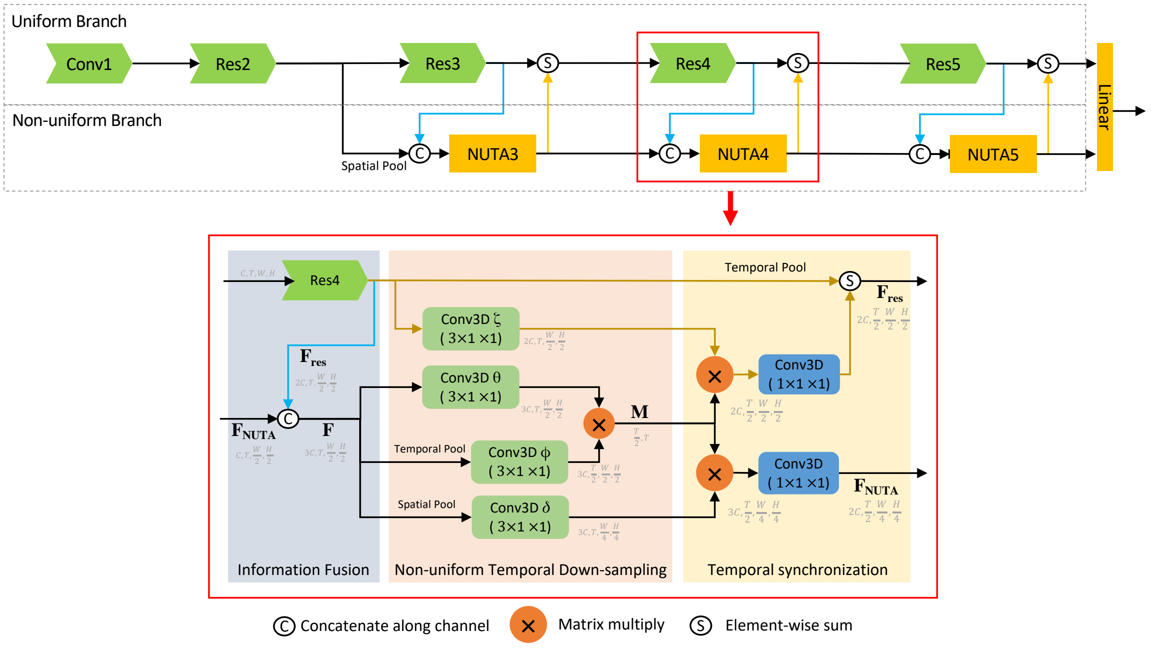
Uniform Branch
Conv1
Res2
Res3
Res4
Res5
Linear
Non-uniform Branch
Spatial Pool
NUTA3
NUTA4
NUTA5
C,T,W,H
Res4
Temporal Pool
Conv3D ζ (3×1 ×1)
$2C,T,\frac{W}{2},\frac{H}{2}$
$\mathbf{F}_{\mathbf{res}}$
$2C,\frac{T}{2},\frac{W}{2},\frac{H}{2}$
Conv3D θ (3×1 ×1)
$3C,T,\frac{W}{2},\frac{H}{2}$
$\mathbf{F}_{\mathbf{res}}$
$2C,T,\frac{W}{2},\frac{H}{2}$
M
$\frac{T}{2},T$
Conv3D (1×1 ×1)
$2C,\frac{T}{2},\frac{W}{2},\frac{H}{2}$
$\mathbf{F}_{\mathbf{NUTA}}$
$C,T,\frac{W}{2},\frac{H}{2}$
F
$3C,T,\frac{W}{2},\frac{H}{2}$
Temporal Pool
Conv3D ϕ (3×1 ×1)
$3C,\frac{T}{2},\frac{W}{2},\frac{H}{2}$
Conv3D (1×1 ×1)
$\mathbf{F}_{\mathbf{NUTA}}$
$3C,\frac{T}{2},\frac{W}{4},\frac{H}{4}$
$2C,\frac{T}{2},\frac{W}{4},\frac{H}{4}$
Spatial Pool
Conv3D δ (3×1 ×1)
$3C,T,\frac{W}{4},\frac{H}{4}$
Information Fusion
Non-uniform Temporal Down-sampling
Temporal synchronization
C Concatenate along channel
× Matrix multiply
S Element-wise sum

Figure 2: An overview of proposed NUTA network: a two-branch design (top). A detailed view of our NUTA module and its interaction with uniform branch is show on the bottom. ResN denotes the N^{th} residual block in the I3D backbone.

can be intuitively understood as applying the same temporal projection to the uniform branch feature $\mathbf{F}_{\mathbf{res}}$, so that feature is synchronized with clip-level feature from the non-uniform branch. Note that although we apply the same temporal projection map to the feature from uniform branch, the following 3D resnet layers will still generate the local features. The temporal synchronization removes the non-informative features from uniform branch, which helps improve the action recognition performance and reduces the computation cost (see ablation study section).

3.4. Information Fusion

The temporal synchronization handles the synchronization and combination of the non-uniform features with the uniform branch, however it doesn't encode the spatial information learned in the uniform branch into the non-uniform branch. To address this, we fuse the uniform branch feature from res-stage n and non-uniform feature from NUTA-stage $n-1$ and feed them to NUTA-stage n as follows:

$$\mathbf{F}_n = f[\mathbf{F}_{\mathbf{NUTA}n-1}, \mathbf{F}_{\mathbf{res}n}] \tag{4}$$

where $\mathbf{F}_n$ stands for the fused feature at n_{th} stage. $\mathbf{F}_{\mathbf{NUTA}n-1}$ denotes the feature from non-uniform branch (NUTA module) and $\mathbf{F}_{\mathbf{res}n}$ is the feature from the uniform branch (ResNet feature backbone). We consider the following three fusion strategies f: (1). Non-local fusion (f_n): where we use $\mathbf{F}_{\mathbf{NUTA}n-1}$ as query to pull information from $\mathbf{F}_{\mathbf{res}n}$. (2). Feature sum-up ($f_s$): where we apply the 1D convolution to $\mathbf{F}_{\mathbf{NUTA}n-1}$ and apply the pixel wise addition to $\mathbf{F}_{\mathbf{res}n}$. (3). Channel concatenate (f_c): where we concatenate $\mathbf{F}_{\mathbf{res}n}$ with $\mathbf{F}_{\mathbf{NUTA}}$ along channel dimension. Our results show that f_c gives best performance at lowest computational cost. Note that the initial $\mathbf{F}_{\mathbf{NUTA}}$ is generated by spatially down-sampling the uniform branch feature as shown in Figure 2.

3.5. Implementation Details

We initialize the uniform branch with Res2D or I3D backbone networks using ImageNet pre-trained weights. Our experiments show that the training process converges faster if the uniform branch is initialized with Kinetics 400 pre-trained weights. For the other components, we used kaiming initialization [12]. The model is trained on 8 machines with 8 GPUs on each machine (64 Tesla V100 GPUs in total). The batch size is set to 8 per GPU without using the synchronized batch normalization. If the batch size is smaller than 8 per GPU, we train the model with synchronized batch normalization.

All models are trained for 180 epochs. We use an initial learning rate of 0.4 (equivalent to 0.05 on a single instance) and weights decay of 1e-4. Learning rate is dropped by scale of 10 at epoch 60, 120 and 150. To avoid over-fitting, we apply the commonly used data augmentation techniques including: random resized crop (i.e., with short side randomly resized to [256, 320] and random crops of 224 following [30, 36]), random temporal sampling (i.e., we keep the original video frame rate, when a video has less frames than needed, we pad the missing frames with last frame) and random horizontal flipping. Besides data augmentation, we also use dropout [13] before the final linear layer with a dropout ratio of 0.6.

4. Experimental Results

4.1. Dataset

We test our method on four of the most widely used datasets:

Kinetics 400 [4] consists of approximately 240k training and 20k validation videos trimmed to 10 seconds from 400 human action categories. Similarly to other works, we report top-1 classification accuracy on the validation set.

Kinetics 700 [3] contains approximately 650k video clips that covers 700 human action classes. Each action class has at least 600 video clips. Each clip is human annotated with a single action class and lasts around 10s. We report top-1 classification accuracy on the validation set.

Something-Something V2 [11] dataset consists of 174 actions and contains 220,847 videos. Following other works, we report top-1 classification accuracy on the validation set. Something-Something dataset is fairly different from the other three datasets as it does not depict people performing common activities but instead manipulations of one or two objects such as opening something, covering something with something, moving something behind something. This dataset in particular requires strong temporal modeling as many activities cannot be simply inferred based on spatial features.

Charades [28] has about 9.8k training videos and 1.8k validation videos in 157 classes in a multi-label classification setting with longer activities spanning about 30 seconds on average. Performance is measured in mean Average Precision (mAP).

4.2. Comparison to State-of-the-art

4.2.1 Kinetics 400

Following previous works, at inference we run 30-view (10 temporal crops and 3 spatial crops) through our network and average their predictions to procure the final classification score. We report results on the validation set of Kinetics 400 in Table 1. We report the top 1 accuracy and GFLOPs (Giga Floating-Point Operations) required to compute results on one view.

Our method achieves comparable performance to state-of-the-art methods with fewer FLOPs and notably outperforms other methods that have a similar goal of answering where to focus the feature extraction across the video clip. In comparison to the non-local model that emphasizes long-term temporal modeling [36], our network achieves 1.2% higher accuracy with roughly half of the computation. Looking at our I3D-50 variant, we outperform the most recent TEA network [20] by 0.7% at roughly same FLOPs using 16 frames input. Compared to state-of-the-art networks [8, 42], our model achieves similar performance with 30% less FLOPs compared with TPN [42]. The results demonstrate that dropping non-informative frames is an efficient way for temporal feature aggregation in comparison to fusing features at multiple different scales. The proposed NUTA network boosts the performance of an I3D network by 3.2% but only increases computation by about 8%, which demonstrates the proposed network architecture is computationally efficient.

Model	Input	GFLOPs	Top1
R2D50 [36]	32 × 2	-	69.9
R2D101 [36]	32 × 2	-	71.3
R2D50-NL [36]	32 × 2	-	73.5
TSM [22]	8	33	74.1
NUTA R2D-50	32 × 2	54	74.2
NUTA R2D-101	32 × 2	66	75.5
I3D50* [2]	32 × 2	33	74.0
I3D101* [2]	32 × 2	58	77.4
I3D50 [42]	32 × 2	168	75.7
NL50 [36]	32 × 2	282	76.5
NL101 [36]	32 × 2	544	77.7
TEA50 [20]	16 × 2	70	76.1
CIDC [18]	32 × 2	101	75.5
FG 152 [26]	32 × 2	-	78.8
SF50 [8]	32 × 2	66	77.0
SF101 [8]	32 × 2	106	77.5
TPN-50 [42]	32 × 2	256	77.7
TPN-101 [42]	32 × 2	458	78.9
X3D-L [7]	16 × 5	24.8	77.5
NUTA network I3D-50*	32 × 2	50	76.2
NUTA network I3D-50	16 × 4	98	76.8
NUTA network I3D-50	32 × 2	197	77.2
NUTA network I3D-101*	32 × 2	111	77.6
NUTA network I3D-101	16 × 4	186	78.3
NUTA network I3D-101	32 × 2	372	78.9

Table 1: Results on Kinetics-400 dataset. We report top 1(%) on the validation set. I3D and I3D* stand for the I3D network w/o and with temporal down sampling respectively [36]. The 'Input' column indicates what frames of the 64 frame clip are actually sent to the network. $n \times \tau$ input indicates we feed n frames to the network sampled every τ frames.

NUTA is generalizable to different backbones. From the table we show that NUTA network works well with both the I3D variant that includes temporal down-sampling [36] and the one that has no temporal down-sampling [42]. Of the methods in Table 1, X3D [7], which is based on neural architecture search, arguably has the best trade-off between performance and cost. Our NUTA network is fully compatible with it and other 3D convolution based backbones as well.

The NUTA module also works with 2D backbones. The results in the top rows of Table 1 show that our model is able

Model	pre-train	FLOPs	Top1
I3D50 [3]	N/A	N/A	58.7
SF101-NL 8 × 8	K4&K6	115G	70.6
SF101-NL 16 × 8	K4&K6	234G	71.0
NUTA network I3D-50 (32 × 2)	K400	197G	68.9
NUTA network I3D-101 (16 × 4)	K400	186G	69.5
NUTA network I3D-101 (32 × 2)	K400	372G	70.3

Table 2: Results on Kinetics-700 dataset. We report top 1 accuracy (%) on validation set. K4&K6 stand for Kinetics 400 and Kinetics 600. SF-NL 101 stands for slowfast network with non-local block [8]. $n \times \tau$ indicates we feed n frames to the network sampled every τ frames.

to outperform the recently proposed 2D TSM [22] network with the similar computational cost, which demonstrates that the non-uniform temporal information aggregation is effective on both 2D and 3D feature extraction pipelines.

4.2.2 Kinetics 700

Kinetics 700 [3] is the latest Kinetics dataset for video classification. We follow the same 30-view evaluation protocol as used in Kinetics 400 and report the FLOPs and top1 accuracy.

Our experiments show a consistent performance trend on Kinetics 700 dataset. As listed in Table 2, our model achieves comparable performance to slowfast [8], and significantly outperforms the baseline I3D by 10%. Note that the slowfast model is pre-trained on both Kinetics 400 and 600 while our model is only pre-trained on Kinetics 400, which may explain the small performance gap between our model and slowfast with non-local [8].

4.2.3 Something-Something V2

Something-something dataset is unique in that it shows people manipulating objects, rather than performing common actions. We believe that distinguishing those manipulation actions needs more motion context than other datasets so it's a good dataset to validate the effectiveness of temporal modelling. Following previous works [22]. during inference, we use the center crop of size 224 × 224 from 8 segments to compute the classification score on the validation set.

As the results in Table 3 show, our model achieves state-of-the-art comparable recognition performance (single model with RGB input) on something-something v2 dataset with 63.0% top1 accuracy. Our NUTA network is able to outperform I3D and I3D based approaches [39, 18] with lower computational cost. It is worth mentioning that our NUTA network generalizes well to different datasets. Previous works have demonstrated that I3D is not as effective in modeling the temporal relations [22, 18]. The state-of-the-art methods on Kinetics e.g. TPN [42] and slowfast [42] must either use the TSM [22] backbone or the multigrid training trick to achieve high performance on the Something-something dataset, showing that for Something-something dataset a specialize adaptation is usually required to transfer a high performing Kinetics model over to the task. We apply our NUTA method directly to the unmodified I3D backbone and are able to get a 11.5% performance boost over the I3D baseline. We believe this results demonstrate that our selection of informative features is not only useful in improving performance, it generalizes more easily to different domains of video understanding.

Model	Init.	FLOPs	Top1
I3D50	K400	33G	50.0
TRN [44]	ImgNet	42G	55.5
CIDC [18]	K400	92G	56.1
TSM [22]	K400	33G	59.1
SF [39]	K400	66G	60.9
SF (multigrid) [39]	K400	66G	61.2
TPN [42]	K400	56G	62.0
NUTA network I3D-50	K400	49G	61.5
NUTA network I3D-101	K400	98G	62.1
NUTA network I3D-101	K700	98G	63.0

Table 3: Results on Something-something V2 validation set. The results are generated by taking the center crop of 1 clip/video [22] as input to the network.

4.2.4 Charades

Charades is a dataset [28] of longer duration video annotated for a multi-label action classification problem. Table 4 shows our model achieves very competitive recognition accuracy that is comparable to state-of-the-art methods while at significantly lower computational cost. Compared to both slowfast-101 and slowfast-101 with non-local [8], we achieve slightly lower performance (41.2 mAP) but with 18% fewer FLOPs. We achieve state-of-the-art performance when we initialize the model with K700 pre-trained weights, but still at much lower computational cost comparing with previous methods [38, 8].

4.3. Ablation Study

4.3.1 NUTA vs. uniform temporal aggregation

We compare the performance of an I3D model with temporal down-sampling (temporal pooling and striding) [36], an I3D without temporal down-sampling [42] and an I3D with our proposed NUTA network (Table 5(a)). The results show that the I3D without temporal down-sampling outperforms the I3D with temporal downsampling. We believe this is because it is able to maintain more temporal information.

Model	Init.	FLOPs	mAP
Non-local [36]	K400	544G	37.5
STRG [37]	K400	N/A	39.7
LFB-NL [38]	K400	529G	42.5
SF [8]	K400	213G	42.1
SF-NL [8]	K400	234G	42.5
X3D-XL [7]	K400	48.4G	43.4
NUTA network 101	K400	186G	41.2
NUTA network 101	K700	186G	43.1

Table 4: Results on Charades dataset. The results are generated from an input of 32 frames sampled 1 out of 4 frames.

Our NUTA network is able to further outperform I3D without temporal down-sampling by an additional 1.5%, which shows that non-uniformly aggregating the informative temporal information helps with the action understanding. Note that the NUTA models only add roughly 10% additional GLOPs since the NUTA modules perform their own temporal down-sampling, significantly reducing its computational overhead.

4.3.2 NUTA network branch analysis

To understand the feature from both the uniform and non-uniform branches we look at different configurations of our network using the I3D backbone as a starting point. First we take the baseline by using the I3D backbone as the uniform branch without adopting an non-uniform branch. Then we construct a NUTA network by attaching our NUTA modules to form a non-uniform branch but still perform classification from only the features coming out of the uniform branch (I3D features). Such setup outperforms the baseline by 0.7% (Table 5(b)), and we conjecture this is because the features from uniform branch utilize the temporal mapping learned by the non-uniform branch from NUTA module to drop non-informative frames. Adding the features from non-uniform branch classification further improves the Top1 accuracy by 0.9%, showing that those two branches learn features that are complementary for better action recognition.

4.3.3 Frame rate

We test the NUTA network at different frame-rates following previous papers [8, 36, 42]. Our results (Table 5(c)) shows that dense sampling gives a better performance but requires significantly higher computational cost. Our method particularly benefits from more densely-sampled frames as this prevents information loss at the early stages of the network, thus giving our NUTA units sufficient information for temporal feature aggregation.

4.3.4 Frame resolution

Table 5(d) compares our NUTA network performance across different input resolutions. Generally a larger input resolution leads to a better performance, but will significantly increase the computational cost. We used 256^2 for all other experiments and GFLOPs estimation.

4.3.5 NUTA configuration

We study the impact of number of groups used in the 3D convolutions (equation 1) in our NUTA module. Our results (Table 5(e)) show that 64 groups give the best performance.

4.3.6 Number of NUTA units

Table 5(f) compares including (or not including) different NUTA stages. The results show that adding more NUTA stages does not neccessarily lead to better performance. Adding NUTA stage 4 and 5 gives the best performance (Table 5(f)). A possible explanation is that the NUTA module aggregates features by focusing on important temporal instances but stacking too many NUTA modules will compress feature too aggressively (since each NUTA will down-sample temporal dimension by 2).

4.4. Error Analysis

We perform error analysis on the Kinetics 400 dataset using NUTA model. In Table 6, we show the 5 action classes that are most positively and negatively impacted. We observe that our model improves the recognition performance for actions that exhibit significant motion and cannot be easily recognized by certain object, e.g. “skipping rope, air drumming, dunking basketball (because there are other activities related to basketball), whereas the model is confused for those actions that involves less obvious motion or can be easily recognized by certain objects, e.g. “unboxing is always related to box and garbage collecting is often related to garbage truck”. Given that the proposed model learns clip-level temporal information, it’s easier for the model to differentiate actions that exhibit large motions. We also visually demonstrate that the NUTA is able to better focus on the action related feature compared with 2D resnet, which shows the temporal branch and information exchange layers provide useful temporal information for classification. We also explore how our model performs for several challenging activities that share similar objects but different motion patterns. Our proposed NUTA network improved the activity recognition accuracy of 10 out of 18 activities related to balls (e.g. dunking basketball +16%, catching or throwing baseball +15%). 8 out of 13 dancing related activities get improved by NUTA (e.g. salsa dancing

Model	GFLOPs	Top1
I3D* [36]	33	74.0
I3D [42]	168	75.7
NUTA network I3D	197	77.2

(a) Comparison between I3D with and without temporal down-sampling, and our NUTA network

Model	Top1
I3D (backbone)	75.7
+ NUTA (only I3D features)	76.3
+ I3D and NUTA features	77.2

(b) NUTA network performance with different feature extraction strategies

Input	GFLOPs	Top1
8 × 8	49	75.5
16 × 4	98	76.8
32 × 2	197	77.2

(c) Model performance with different input clip lengths

Input	GFLOPs	Top1
128^2	52	76.3
256^2	197	77.2
312^2	395	77.6

(d) Model performance with different input frame resolution

Backbone	Number of groups	Top1
I3D*	16	75.6
I3D*	32	76.0
I3D*	64	76.2

(e) Model performance with different NUTA group head configurations

Backbone	Add NTUA at	Top1
I3D	NUTA3+NUTA4+NUTA5	76.7
I3D	NUTA4+NUTA5	77.2
I3D	NUTA5	75.7

(f) Performance comparison of including different NUTA stages

Table 5: Ablation studies on Kinetics 400. We use an I3D-50 backbone. I3D and I3D* stand for the I3D network w/o and with temporal down sampling respectively. The evaluation is performed on 30 views with 32 frame input unless specified.

Table 6: Quantitative analysis on Kinetics-400 dataset. The performance gain is defined as the disparity of the top-1 accuracy between I3D-50 and that of our NUTA network.

Top 5 Activity (+)	Gain	Top 5 Activity (-)	Gain
skipping rope	+26%	unboxing	-38%
air drumming	+24%	tasting food	-34%
massaging head	+19%	zumba	-25%
salsa dancing	+16%	applauding	-25%
dunking basketball	+16%	garbage collecting	-22%

+16%, robot dancing +%10, dancing charleston +9%). And we improved on 3 out of 5 activities related to legs (leg +10%, swinging legs +8%, shaving legs +4%).

5. Visualization

To understand how the proposed NUTA unit works to aggregate temporal features, we visualize the temporal projection matrix (equation 1). We notice that: when the input clip has smooth transition, which indicates the information is roughly uniformly distributed over time, the NUTA unit performs uniform sampling (Figure 3 top row). Most previous works based on 3D convolutions also perform temporal information aggregation in this way. When the frames are highly repetitive or contain only background without useful information for action recognition, the NUTA unit is able to skip the non-informative frames and focus more on the representative features (e.g. Figure 3 middle row, information from first half of the clip is skipped). When the input features have scene changes or contain noises (e.g. transition frames), the temporal mapping generated by NUTA stays to focus on representative information by skipping noises (e.g. Figure 3 bottom row, the NUTA is able to focus on the start and end of the video while skipping the middle frames).

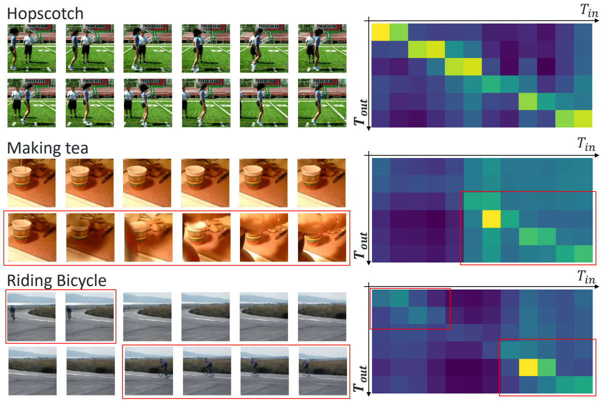

Figure 3: Visualization of the temporal projection map learned by our NUTA module. Our NUTA module performs uniform aggregation when information is smoothly distributed across clip (top clip) but is able to focus on informative temporal segments when there is irrelevant segments in the video (middle and bottom clips).

6. Conclusion

In this paper, we have addressed the "where to extract feature" question by proposing a non-uniform temporal aggregation (NUTA) module, which is able to select the informative features. We have further proposed the two-branch network with a uniform branch that learns local feature and non-uniform branch that learns clip-level features. Our experimental results have shown that the proposed NUTA network achieves state-of-the-art accuracy on four public action recognition datasets. Besides, we have demonstrated how our NUTA network works by visualizing the intermediate temporal projection matrices. One future direction is to apply the proposed NUTA to other tasks (e.g. action detection).

References

[1] Hakan Bilen, Basura Fernando, Efstratios Gavves, Andrea Vedaldi, and Stephen Gould. Dynamic image networks for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3034–3042, 2016.

[2] Joao Carreira. Quo vadis, action recognition. *A new model and the kinetics dataset. CoRR, abs/1705.07750*, 2:3, 2017.

[3] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019.

[4] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, July 2017.

[5] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[6] Wenbin Du, Yali Wang, and Yu Qiao. Recurrent spatial-temporal attention network for action recognition in videos. *IEEE Transactions on Image Processing*, 27(3):1347–1360, 2017.

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

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

[9] Christoph Feichtenhofer, Axel Pinz, and Richard P Wildes. Spatiotemporal multiplier networks for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4768–4777, 2017.

[10] Basura Fernando, Efstratios Gavves, José Oramas, Amir Ghodrati, and Tinne Tuytelaars. Rank pooling for action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):773–787, 2016.

[11] 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 3, 2017.

[12] 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.

[13] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[15] Yueyu Hu, Chunhui Liu, Yanghao Li, and Jiaying Liu. Temporal perceptive network for skeleton-based action recognition. In *BMVC*, 2017.

[16] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[17] Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6232–6242, 2019.

[18] Xinyu Li, Bing Shuai, and Joseph Tighe. Directional temporal modeling for action recognition. *arXiv preprint arXiv:2007.11040*, 2020.

[19] Xinyu Li, Yanyi Zhang, Jianyu Zhang, Moliang Zhou, Shuhong Chen, Yue Gu, Yueyang Chen, Ivan Marsic, Richard A Farneth, and Randall S Burd. Progress estimation and phase detection for sequential processes. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, 1(3):73, 2017.

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

[21] Zhenyang Li, Kirill Gavrilyuk, Efstratios Gavves, Mihir Jain, and Cees GM Snoek. Videolstm convolves, attends and flows for action recognition. *Computer Vision and Image Understanding*, 166:41–50, 2018.

[22] 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*, 2019.

[23] Chunhui Liu, Yanghao Li, Yueyu Hu, and Jiaying Liu. Online action detection and forecast via multitask deep recurrent neural networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1702–1706. IEEE, 2017.

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

[25] Brais Martinez, Davide Modolo, Yuanjun Xiong, and Joseph Tighe. Action recognition with spatial-temporal discriminative filter banks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5482–5491, 2019.

[26] Brais Martinez, Davide Modolo, Yuanjun Xiong, and Joseph Tighe. Action recognition with spatial-temporal discriminative filter banks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[27] 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.

[28] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ivan Laptev, Ali Farhadi, and Abhinav Gupta. Hollywood in

homes: Crowdsourcing data collection for activity understanding. *ArXiv e-prints*, 2016.

[29] 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.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[31] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. Spatio-temporal attention-based lstm networks for 3d action recognition and detection. *IEEE Transactions on image processing*, 27(7):3459–3471, 2018.

[32] Graham W Taylor, Rob Fergus, Yann LeCun, and Christoph Bregler. Convolutional learning of spatio-temporal features. In *European conference on computer vision*, pages 140–153. Springer, 2010.

[33] 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.

[34] 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.

[35] 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.

[36] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

[37] 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.

[38] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 284–293, 2019.

[39] Chao-Yuan Wu, Ross Girshick, Kaiming He, Christoph Feichtenhofer, and Philipp Krahenbuhl. A multigrid method for efficiently training video models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 153–162, 2020.

[40] Wenhao Wu, Dongliang He, Xiao Tan, Shifeng Chen, and Shilei Wen. Multi-agent reinforcement learning based frame sampling for effective untrimmed video recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6222–6231, 2019.

[41] 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.

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

[43] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4694–4702, 2015.

[44] 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.