Diffused Fourier Network for Video Action Segmentation

Borui Jiang
Peking University
Beijing, China
jiangbr@pku.edu.cn

Yadong Mu*
Peking University
Beijing, China
muyadong@gmail.com

ABSTRACT

Video action segmentation aims to densely cast each video frame into a set of pre-defined human action categories. This work proposes a novel model, dubbed as diffused Fourier network (DFN) for video action segmentation. It advances the research frontier by addressing several central bottlenecks in the existing methods for video action segmentation. First, capturing long-range dependence among video frames is known to be crucial for precisely estimating the temporal boundaries for actions. Rather than relying on compute-intensive self-attention modules or stacking multi-rate dilated convolutions as in previous models (*e.g.*, ASFormer), we devise Fourier token mixer over shiftable temporal windows in the video sequence, which harnesses the parameter-free and light-weighted Fast Fourier Transform (FFT) for efficient spectral-temporal feature learning. Essentially, even simple spectral operations (*e.g.*, pointwise product) bring global receptive field across the entire temporal window. The proposed Fourier token mixer thus provides a low-cost alternative for existing practice. Secondly, the results of action segmentation tend to be fragmented, primarily due to the noisy per-frame action likelihood, known as over-segmentation in the literature. Inspired by the recently-proposed diffusion models, we treat over-segments as noises corrupting the true temporal boundaries, and conduct denoising via a recurrent execution of a parameter-sharing module, akin to the backward denoising process in the diffusion models. Comprehensive experiments on three video benchmarks (GTEA, 50salads and Breakfast) have clearly validated that the proposed method can strike an excellent balance between computations / parameter count and accuracy.

CCS CONCEPTS

• **Computing methodologies** → *Video segmentation*; *Machine learning*; **Activity recognition and understanding**.

KEYWORDS

Video action segmentation, Diffusion Models, Fourier token mixer

ACM Reference Format:
Borui Jiang and Yadong Mu. 2023. Diffused Fourier Network for Video Action Segmentation. In *Proceedings of 31th ACM International Conference on Multimedia (MM '23).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

*Corresponding author.

MM '23, October 29, 2023 – November 3, 2023, Ottawa, Canada

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

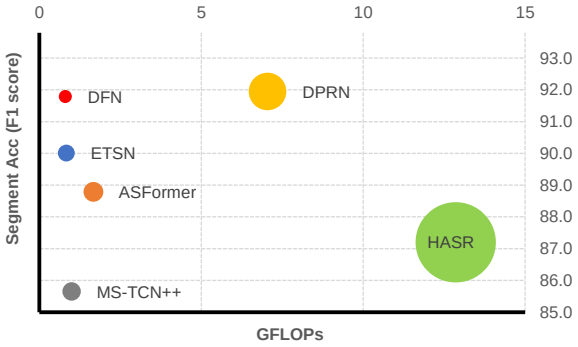

Figure 1: The accuracy-complexity trade-offs for various video action segmentation models. The ones closer to the upper-left and with smaller radius (*i.e.*, fewer parameters) are more favored. Our proposed model is diffused Fourier network (DFN).

1 INTRODUCTION

Automatically segmenting human actions in video sequences is essential for many applications, such as video surveillance [11], robotics [20, 61], and future prediction for sequential data [47]. However, this task remains challenging due to the tremendous variations in the visual appearance and temporal dynamics of videos. Developing fast and accurate methods for temporal action segmentation is still an active research topic in computer vision.

In recent years, multi-stage dilated convolutional networks [15, 17, 37, 45] with non-local interactions such as self-attention [62, 67], hierarchical representations [1, 54], grammar [47] or graph structures [18, 26, 70] have shown promising results in temporal action segmentation. However, most of the state-of-the-art methods suffer from high computational cost (*e.g.*, quadratic time complexity of self-attention modules) or diminishing returns of adding more learnable parameters (*e.g.*, stacked dilated convolutional layers), which limits their practical applications. As such, it is strongly needed for developing light-weight backbone networks that strike a trade-off between computational cost and accuracy, particularly for long-duration or high-frame-rate videos. Figure 1 calibrates such trade-offs for a variety of related models.

We here propose a novel model, dubbed as diffused Fourier network (DFN) for video action segmentation. The proposed model is inspired by recent development along both spectral neural networks and diffusion-based generative models. Figure 2 shows the architectural design of DFN. As seen, one of the defining features of DFN is the Fourier-based token mixer. It efficiently captures

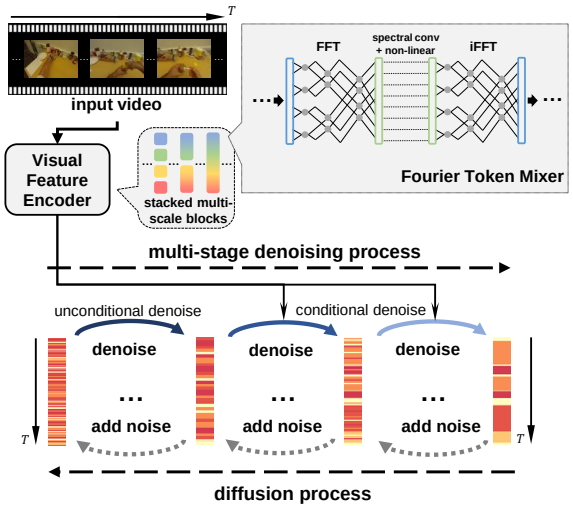

Figure 2: The pipeline of diffused Fourier network (DFN). It adopts stacked multi-scale blocks and Fourier token mixer to efficiently extract visual features, and a multi-stage denoising process to predict the labels for all video segments.

long-term temporal dependencies along different spectral frequencies by taking advantage of the Convolution theorem [29]. Specifically, we use a sliding window to divide the video sequence into non-overlapping windows and apply a Fourier transform to each window. The Fourier coefficients are used as tokens that can be mixed by spectral convolution and non-linear activation, rather than standard compute-intensive self-attentions. Compared to the self-attention module, the Fourier-based token mixer requires significantly fewer parameters and computations, making it more practical for real-world applications.

Another distinguishing trait of DFN is the utilization of multi-stage diffusion-based blocks. Over-segmentation is a notorious challenge in video action segmentation, referring to the undesired fragmented action segments primarily caused by noisy frame-level likelihood across action categories. DFN progressively refines the initial over-segmentated predictions via a denoising process, guided by the diffusion time embeddings and the diffusion coefficient parameters. This allows the model to focus on local cues and suppress the noise in action predictions, resulting in smoother and more accurate action boundaries. In addition, the entire diffusion process was divided into multiple stages, either conditioned or not by the visual inputs, as shown in Figure 2. The unconditional diffusion stage forces the model to seek cues from the prior regularity from general human action sequences, while the conditional diffusion stages further improves over-segmented prediction through denoising under visual information. Additionally, the multi-stage diffusion models are able to share parameters and raw visual features across time steps, which can further help reduce the number of parameters and computational complexity of the model, making it more efficient and scalable.

We empirically evaluate the proposed method on three standard datasets for video action segmentation, including GTEA [16], 50salads [58] and Breakfast [31]. The experimental results consistently show that, while demanding significantly lower computational cost and fewer parameters, the proposed DFN can still achieve superior or comparable performances compared with its state-of-the-art competitors. Comprehensive ablation investigations also validate the effectiveness of the multi-stage diffusion module for mitigating over-segments, and the Fourier-based token mixer in capturing long-range temporal dependencies. All above make DFN a promising solution for real-world applications that require efficient and accurate human action segmentation.

2 RELATED WORKS

Video action segmentation. Early approaches for action segmentation are based on detecting motion boundaries [42, 50], sliding windows [12, 69], or grammar representations [5, 41, 44, 55, 64] such as hidden Markov models (HMM) [48] and conditional random fields (CRF) [59]. Recently, deep neural networks with temporal convolutions [15, 32, 45] have made great success on densely labeling general human activities. Meanwhile, methods exploring different high-level representations, such as human-designed rules [1, 27], self-attention mechanism [62, 67], sequence translation [4] or graph-based embeddings [18, 26, 70], also perform remarkably well. However, considering the complex high-order temporal correlation in videos, modern models for video action segmentation continue to become more heavy-weighted in terms of parameters and time complexity. In particular, the over-segmentation issue is usually solved by stacking many refining modules [37, 45] or through dynamic smoothing [63], incurring the increase of model parameters and computational cost. Some prior works [4] directly predict the action sequence to avoid using inefficient refinement modules, while ETSN [38] tackles this issue by efficiently capturing semantic features and unsupervised refinement. Different from above methods, we provide a low-cost alternative by computing in some spectral domain, and alleviate the over-segmentation errors by multi-stage diffusion models.

Denoising diffusion models. Diffusion models [23] initially emerged in vision tasks for content generation, and more recently gain its popularity in. some other non-generative vision tasks, such as image classification [71], panoptic segmentation [2, 9], object detection [8], instance segmentation [19], and video recognition [25]. Different from the feedforward-style data flow in traditional deep neural networks, the stochastic denoising process in diffusion models entails a diversity-encouraging generative model. Aforementioned methods leverage conditional (rather than the original unconditional variant) diffusion process [57] to gradually get rid of the noise that contaminates the data, guided by some reference information (*e.g.*, the encoding of the input image or text-enforced target semantics). Inspired by the iterative diffusion frameworks in image super-resolution [24, 52], we propose a cascaded diffusion network that combines both unconditional and conditional diffusion stages for action segmentation, which regards over-segmentation errors as undesired noises and accordingly reformulates the problem.

Spectral neural networks. Learning feature representation in some spectral domain (*e.g.*, those induced via Fourier or wavelet transforms) is recently attracting much research enthusiasm owing to the non-local essence inherited from the spectral transform. Existing works have explored the hybrid use of spatial, temporal

and spectral information [10, 34, 65, 66]. For example, Fast Fourier Convolution [10] proposed to mix the spatial-temporal and spectral weights to efficiently capture global and local dependencies in parallel. AutoFormer [66] devised an auto-correlation mechanism to capture the series-wise temporal dependencies based on the learned periods. FNet [34] provided an "efficient transformer" of simpler token mixing mechanisms by constantly switching between temporal and spectral domain. TimesNet [65] extended the analysis of temporal variations into the 2D space under the observation of multi-periodicity in time series. In this work, we propose Fourier token mixer to jointly enhance the temporal and spectral features sequentially, which supposedly introduce such inductive bias that can improve the performance of temporal action segmentation.

3 THE PROPOSED METHOD

3.1 Preliminaries

We denote a RGB video sequence as $\mathbf{V} = \{\mathbf{v}_t\}_{t=1}^{T} \in \mathbb{R}^{H\times W\times 3\times T}$, where $\mathbf{v}_t$ represents the video frame at time t, with a spatial resolution $H \times W$. Let $\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^{T} \in \mathbb{R}^{C\times T}$ be the sequence of feature maps extracted from $\mathbf{V}$ using a pre-trained backbone network(*e.g.*, I3D [6] or ViT [13]), where C is the dimension of features. We define the ground-truth category of $\mathbf{V}$ as a sequence $\mathbf{Y}_0 = \{\mathbf{y}_t\}_{t=1}^{T} \in \{0,1\}^{K\times T}$, where $\mathbf{y}_t$ represents a one-hot vector at time t and K indicates the total number of categories.

Diffusion models [23] aim to generate samples from Gaussian noises via bi-directional processes: diffusion process (add noises) and denoising process (remove noises). Specifically, diffusion process adds Gaussian noises step by step from the real sample $s_0 \sim q(s)$ to generate $s_1, \ldots, s_M$, which can be formulated as:

$$q(s_i|s_{i-1}) = \mathcal{N}(s_i; \sqrt{1-\beta_i}s_{i-1}, \beta_i\mathbf{I}), \tag{1}$$

where $\mathcal{N}$ denotes the normal distribution and β_i is user-settable variance schedule. And

$$q(s_{1:M}|s_0) = \prod_{i=1}^{M} q(s_i|s_{i-1}). \tag{2}$$

On the other hand, the denoising process removes the noises step by step from a pure Gaussian noise $s_M \sim \mathcal{N}(0, \mathbf{I})$ to generate s_0 by a parameterized distribution p_θ, which can be formulated as:

$$p_\theta(s_{0:M}) := p(s_M) \prod_{i=1}^{M} p_\theta(s_{i-1}|s_i), \tag{3}$$

with the variational assumption below:

$$p_\theta(s_{i-1}|s_i) = \mathcal{N}(s_{i-1}; \mu_\theta(s_i, i), \Sigma_\theta(s_i, i)). \tag{4}$$

The seminal DDPM model [23] proposed to fix $\Sigma_\theta(s_i, i)$ to a constant value $\sigma_i^2\mathbf{I}$ determined by β_i, and rewrite $\mu_\theta(s_i, i)$ as a function of the estimated noise $z_\theta(s_i, i)$, or a function of $\hat{s}_\theta(s_i, i) \approx s_0$. Take $\hat{s}_\theta(s_i, i)$ as an example:

$$\mu_\theta(s_i, i) = \frac{\sqrt{\alpha_i}(1-\overline{\alpha}_{i-1})s_i + \sqrt{\overline{\alpha}_i}(1-\alpha_{i-1})\hat{s}_\theta(s_i, i)}{1-\overline{\alpha}_i}, \tag{5}$$

where $\alpha_i = 1-\beta_i$, $\overline{\alpha}_i = \prod_{k=1}^{i} \alpha_k$ and $\hat{s}_\theta$ is an output of a neural network, such as RNN [43], LSTM [53] or U-Net [49]. During inference, s_i is denoised by predicted $\hat{s}_\theta$ step-by-step and finally generate s_0. As described above, the parameterized distribution p_θ only generate

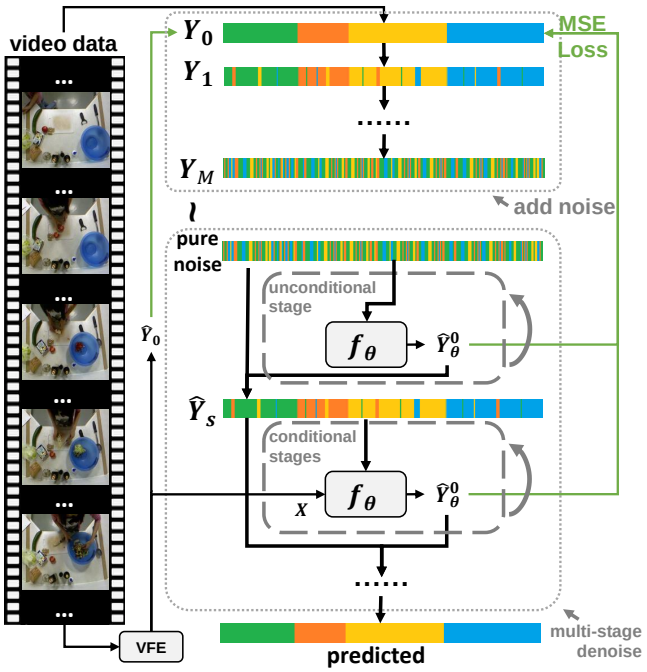

Figure 3: Illustration of the proposed multi-stage diffusion model pipeline. The overall framework contains an ordinary diffusion process and a cascaded multi-stage denoising process. There are two heterogeneous ingredients during denoising: an unconditional stage and several conditional stages. The unconditional stage denoises from pure noise using the priors from data, while the conditional stages do the rest job conditioned on the output X of Visual Feature Encoder (VFE). See the main text for more details.

s_{i-1} by the current noised sample s_i and step index i at each step. Moreover, several vision tasks such as super-resolution [24] or text-image generation [51] are also interested in learning conditional distribution $p_\theta(s|x)$.

3.2 Cascaded Multi-Stage Diffusion Networks

As shown in Figure 3, a cascaded multi-stage diffusion (MSD) temporal network is designed to efficiently predict action segmentation sequences. The first stage is an unconditional denoising process that learns a prior distribution of actions over the full videos from the training data. The remaining stages use the results generated in previous stage as a reference template and adds visual features as extra conditions for further denoising to generate the final segmentation. Similar to the traditional diffusion models, the goal of MSD is also to recover the input from the noise: it regards $\mathbf{Y}_0$ as s_0 and performs both the diffusion process and the denoising process. However, the Gaussian noises can not be applied to discrete one-hot vector sequence $\mathbf{Y}_0$. To map the discrete vectors into the real field, we choose the modified inverse function g of softmax, which is formulated as:

$$g(\mathbf{x}) = (1-\mathbf{x})log(\mathbf{x}+\epsilon) + \mathbf{x}log(\mathbf{x}-(K-1)\epsilon), \ \mathbf{x} \in \{0,1\}^K, \tag{6}$$

where ϵ is a hyper-parameter to prevent value explosion. The mapped $\mathbf{Y}$ is also known as "logits". For brevity, we omit all the g in the formula below. Then the diffusion process can be formulated

as:

$$q(\mathbf{Y}_i|\mathbf{Y}_{i-1}) = \mathcal{N}(\mathbf{Y}_i; \sqrt{\alpha_i}\mathbf{Y}_{i-1}, (1-\alpha_i)\mathbf{I}), \tag{7}$$

where $\alpha_i = 1 - \beta_i$ and

$$q(\mathbf{Y}_{1:M}|\mathbf{Y}_0) = \prod_{i=1}^{M} q(\mathbf{Y}_i|\mathbf{Y}_{i-1}), \tag{8}$$

which is the same as Equation 1. The denoising process of MSD also uses parameterized p_θ to estimate distribution, but it is slightly different from the traditional diffusion models. Specifically, MSD adopts cascaded multi-stage denoising. The first stage does not read visual features as its conditions, while the remaining stages are conditioned on the visual features. In implementation, we achieve this by setting the visual information of each step as:

$$\mathbf{X}_i = \begin{cases} \mathbf{X}, & i \le M', \\ \varnothing, & i > M', \end{cases} \tag{9}$$

where $M' \in \{0, 1, \ldots, M-1\}$ is a hyper-parameter to control the number of steps on unconditional stages. Formally, we have the definition of parameterized distribution p_θ in MSD:

$$p_\theta(\mathbf{Y}_{0:M}|\mathbf{X}) \coloneqq p(\mathbf{Y}_M) \prod_{i=1}^{M} p_\theta(\mathbf{Y}_{i-1}|\mathbf{Y}_i, \mathbf{X}_i), \tag{10}$$

where

$$p_\theta(\mathbf{Y}_{i-1}|\mathbf{Y}_i, \mathbf{X}_i) = \mathcal{N}(\mathbf{Y}_{i-1}; \mu_\theta(\mathbf{Y}_i, i, \mathbf{X}_i), \sigma_i^2\mathbf{I}), \tag{11}$$

and $\mu_\theta(\mathbf{Y}_i, i, \mathbf{X}_i)$ is rewritten as a function of the estimated $\mathbf{Y}_0$, which is formulated as:

$$\mu_\theta(\mathbf{Y}_i, i, \mathbf{X}_i) = \frac{\sqrt{\alpha_i}(1-\overline{\alpha}_{i-1})\mathbf{Y}_i + \sqrt{\overline{\alpha}_i}(1-\alpha_{i-1})\hat{\mathbf{Y}}_\theta(\mathbf{Y}_i, i, \mathbf{X}_i)}{1-\overline{\alpha}_i}. \tag{12}$$

In above formula, $\overline{\alpha}_i = \prod_{k=1}^{i} \alpha_k$ and $\hat{\mathbf{Y}}_\theta(\mathbf{Y}_i, i, \mathbf{X}_i)$ is generated by a simplified temporal neural network $f(\theta; \mathbf{Y}_i, i, \mathbf{X}_i)$, which is optimized by a mean squared loss:

$$\mathcal{L} = ||f(\theta; \mathbf{Y}_i, i, \mathbf{X}_i) - \mathbf{Y}_0||^2. \tag{13}$$

For accelerating, the denoising diffusion implicit model (DDIM) [56] is also used during training and inference.

The idea underlying MSD is to divide the denoising process of the diffusion model into several stages. The first stage of the network starts with pure random Gaussian noise. Since there is no visual information as a condition, this part of the parameters is forced to learn the prior knowledge from the training data, and also avoids confusion in visual features, such as motion blur or over-fitting features, etc. In the remaining stages, MSD refines the action segmentation results generated by the preivous stage by incorporating visual features as conditions. With the assistance of visual features, the network moves towards a more accurate estimate of the action segmentation sequence during the diffusion process. Moreover, the segmentation becomes progressively smoother and closer to the true distribution as more information is propagated by multiple stages. It has been proved in previous works [37, 45, 67] that such structure of multi-stage refinement can effectively solve the problem of over-segmentation. However, such refinement modules often suffer from overfitting or the input distribution misalignment, thus networks designed to be recurrent by simply stacking or sharing their weights usually perform poorly, which is discussed in the Section 4. Unlike these networks, diffusion models regards such refinement as multi-step denoising process, which can save a lot of parameters while continuously refining and smoothing the generated segmentation results.

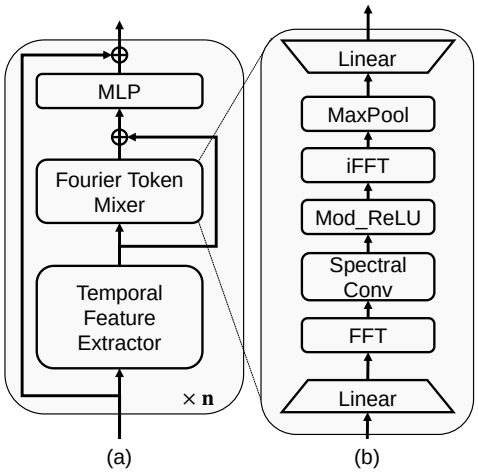

Figure 4: Architectural design of neural blocks (a) with Fourier token mixer (b). In (a), a basic block contains a temporal feature extractor, a Fourier token mixer, and a multi-layer perceptron (MLP), which aggregate temporal and spectral features sequentially with some residual connections. In (b): Temporal features are firstly transformed tothe spectral domain through linear dimensional reduction and window-based Fast Fourier Transformation (FFT). Then, the spectral convolution with `ModReLU` activation mixes all tokens of each sliding window. Finally, after the inverse FFT and MaxPooling, the features are restored to the original dimension by another linear transformation.

3.3 Fourier Token Mixer

The self-attention module in transformer models is computationally expensive, making it difficult to be optimized on long sequences. Previous works [34, 39, 68] have made great efforts on exploring the alternatives of the self-attention mechanism. These methods experiment with various light-weighted operators to mix tokens instead. However, like traditional transformers, all of them are designed for fixed-length inputs, which limits their ability to capture multi-scale dependencies in long-duration videos. For this reason, we here propose to divide the whole sequence into non-overlapped multi-scale windows and adopt efficient operations (*e.g.*, pointwise product) in some spectral domain for capturing multi-scale global dependencies. Different from the traditional self-attention mechanism via sliding window [67], this approach does not face the trade-off between the size of the receptive field and the computational cost, since the larger window sizes of self-attention requiring the exponentially computational costs. We term the proposed new module as Fourier Token Mixer, which is inserted into the pipeline as a neural block, shown in Figure 4.

Formally, supposed that x is the input feature of a module, the temporal feature-extracting module is formulated as:

$$f^t(x) = \sigma_G(\, norm(x) \cdot W^t\,), \tag{14}$$

where *norm* refers to the Instance Normalization [60]. W^t is a temporal dilated convolution [7] with kernel size = 3, σ_G is the `GELU` activation [22]. Like most temporal networks [15], this module

relies on stacking of multi-scale residual blocks to extract long dependencies, while self-attention mechanism makes the model more effective in solving global and local aggregation. However, mining the intrinsic relationships of spectral features is a more effective way to mix tokens. According to the Convolution Theorem [29], updating a single value in the spectral domain globally affects all original data, enabling the Fourier Token Mixer to cope with long temporal dependencies. Compared to self-attention mechanism, Fourier Token Mixer only requires $O(l \log l)$ complexity for mixing l tokens, instead of $O(l^2)$. However, the long time segmentation still needs window-based mechanism to further decrease the computational costs. Inspired by the strategy proposed in SwinTransformer [40], the temporal sequence is divided into non-overlapping windows, while enabling the interaction of all tokens by shifting. We also adopt this strategy and achieve an $O(T \log l)$ complexity for each token mixer layer compared to $O(Tl)$ in ASFormer [67], where l denotes the window size. As shown in Figure 4(b), the Fourier Token Mixer is formulated as:

$$f^s(x) = h(\mathcal{F}^{-1}[\sigma_M(\mathcal{F}(x) \cdot W^s)]), \tag{15}$$

where $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ indicate Fourier transformation and its inversion, respectively. W^s is a group of complex parameters to make inner product in the spectral domain. $h(\cdot)$ is a function for feature aggregation, such as MaxPooling, and σ_M indicates the `ModReLU` activation [3], which is formulated as:

$$\sigma_M = \begin{cases} (|z| + b)\frac{z}{|z|}, & |z| + b \geq 0, \\ \mathbf{0}, & |z| + b < 0. \end{cases} \tag{16}$$

For the sake of conciseness, the linear transformations for dimension reduction and dimension lifting are omitted in Equation 15. Finally, we use a multi-layer perceptron (MLP) to aggregate the global and local information, which is formulated as:

$$f_k(x) = x + \texttt{MLP}(f_k^t(x) + f_k^s(f_k^t(x))), \tag{17}$$

where k indicates the k-th block. It is worth noting that the spectral branch simply use the features from temporal branch, which is different from the aggregating in parallel of Fast Fourier Convolution (FFC) [10]. This is because sequential aggregating mechanism can enable the model to use fewer blocks to obtain larger receptive field. Moreover, this neural block dominated by temporal feature extraction modules involves more inductive bias, which is supposed to be helpful for the temporal action segmentation task. As a result, the Fourier token mixer is able to capture global dependencies with fewer parameters and lower computational complexity compared with the self-attention-based models. Moreover, the multi-scale window size and fusion strategy allow our model to capture the relationship between tokens at different scales and frequencies, making it better suited for the task of temporal action segmentation in long-duration videos. This is crucial, as the prominence of multi-scale actions is a key challenge in long-duration videos.

4 EVALUATIONS

4.1 Experimental Setup

Implementation details. We implement our model using PyTorch [46], and all experiments are executed on a machine with an NVIDIA GeForce GTX Titan X GPU. We use Adam optimizer [30] with a learning rate of 0.0005 and batch size of 1. The whole DFN pipeline contains two parts: the Visual Feature Encoder (VFE) and the Multi-Stage Diffusion (MSD). VFE is a multi-stage temporal network with the stacked Fourier Token Mixer blocks, namely Fourier Network (FN). We adopt totally 5 blocks per stage and set the dilation of temporal convolution to 4^k for each block k, while the window size is $4^{k+1} + 1$ for Fourier token mixer. MSD is a denoising diffusion model with $M = 100$ steps, which are divided into multiple stages during the denoising process. The parameters are shared in the same stage. In function g, ϵ is set to 0.001.

Datasets. The evaluation results are based on 3 popular datasets: Georgia Tech Egocentric Activities (GTEA) [16], 50salads [58] and the Breakfast dataset [31]. The **GTEA** dataset contains 7 types of daily activities, each performed by 4 different subjects with 11 action classes including *background*. GTEA only has meta actions such as *take*, *put* or *open*, etc. Each video contains about 600 to 1000 frames at 15 fps. It will perform official 4-fold cross-validation and report the average results for evaluation. The **50salads** dataset captures 25 people preparing two mixed salads each and contains over 4h of annotated accelerometer and RGB-D video data with 19 action classes including *action start* and *action end*. Each video contains about 20 actions and about 4000 to 9000 frames at 15 fps. In the experiments, only RGB features extracted from I3D networks [6] are used. It will perform official 5-fold cross-validation and report the average results for evaluation. The **Breakfast** dataset is among the largest dataset for action segmentation task, which has 1,712 videos of cooking breakfast in the kitchen environment with a overall duration of 77h. It has 48 different actions including *SIL* and will perform official 4-fold cross-validation and report the average results for evaluation.

Evaluation metrics. For all the dataset, the following evaluation metrics are reported: frame-wise **accuracy**, the segmental **edit score**, and the segmental **F1 scores** at temporal intersection over union (tIoU) thresholds of 0.10, 0.25 and 0.50, denoted by $F1@\{0.10, 0.25, 0.50\}$. The number of the parameters(#) and computational complexity — Giga Floating-point Operations Per Second (GFLOPS) are also discussed in this paper.

4.2 Comparison with State-of-the-art

We compare the proposed cascaded DFN with state-of-the-art methods on three datasets. The detailed results are shown in Table 1. It has two parts: methods in the upper table is mainly based on the I3D [6] features, and the methods in the lower part is based on the Br-prompt [36] features. Since most of the methods is neither open-sourced nor provide specific statistics of parameters or computations, it is difficult to make comparisons with completely identical settings. Therefore, DFN gives priority to keeping the lowest parameter amount and computational complexity compared to all existing methods. As seen, without large model size or lots of computation, DFN still achieves competitive results, which makes it more likely to be the solution for real-world applications requiring efficient and accurate human action segmentation.

4.3 Ablation Studies

Fourier token mixer vs. self-attention: To evaluate the effectiveness of Fourier token mixer, we compare it with the self-attention

Table 1: Evaluation results of action segmentation methods on three datasets. Score in bold indicates the best performance.

	50salads					GTEA					Breakfast				
	F1@{10,25,50}			Edit	Acc	F1@{10,25,50}			Edit	Acc	F1@{10,25,50}			Edit	Acc
DTCN [32]	52.2	47.6	37.4	43.1	59.3			-					-		
ST-CNN [33]	55.9	49.6	37.1	45.9	59.4	58.7	54.4	41.9	-	60.6			-		
TResNet [21]	69.2	65.0	54.4	60.5	66.0	74.1	69.9	57.6	64.4	65.8			-		
TRN [35]	70.2	65.4	56.3	63.7	66.9	77.3	71.3	59.1	72.2	67.8			-		
TDRN [35]	72.9	68.5	57.2	66.0	68.1	79.2	74.4	62.7	74.1	70.1			-		
ED-TCN [32]	68.0	63.9	52.6	59.8	64.7	72.2	69.3	56.0	-	64.0			-		43.3
BCN [63]	82.3	81.3	74.0	74.3	84.4	88.5	87.1	77.3	84.4	79.8	68.7	65.5	55.0	66.2	70.4
ASRF [27]	84.9	83.5	77.3	79.3	84.5	89.4	87.8	79.8	83.7	77.3	74.3	68.9	56.1	72.4	67.6
HASR [1]	86.6	85.7	78.5	81.0	83.9	89.2	87.2	74.8	84.5	76.9	74.7	69.5	57.0	71.9	69.4
ETSN [38]	85.2	83.9	75.4	78.8	82.0	91.1	90.0	77.9	86.2	78.2	74.0	69.0	56.2	70.3	67.8
G2L [17]	80.3	78.0	69.8	73.4	82.2	89.9	87.3	75.8	84.6	78.5	76.3	69.9	54.6	74.5	70.8
MS-TCN [15]	76.3	74.0	64.5	67.9	80.7	85.8	83.4	69.8	79.0	76.3	52.6	48.1	37.9	61.7	66.3
MS-TCN++ [37]	80.7	78.5	70.1	74.3	83.7	88.8	85.7	76.0	83.5	80.1	64.1	58.6	45.9	65.6	67.6
UVAST [4]	89.1	87.6	81.7	83.9	87.4	92.7	91.3	81	92.1	80.2	75.9	70.0	57.2	76.5	66.0
DPRN [45]	87.8	86.3	79.4	82.0	87.2	92.9	92.0	82.9	90.9	82.0	75.6	70.5	57.6	75.1	71.7
EUT [14]	89.2	87.5	81.0	82.9	87.4	88.2	87.2	74.0	83.9	77.0	76.2	71.8	59.8	75.0	74.6
CETNet [62]	87.6	86.5	80.1	81.7	86.9	91.8	91.2	81.3	87.9	80.3	**79.3**	74.3	61.9	**77.8**	74.9
ASFormer [67]	85.1	83.4	76.0	79.6	85.6	90.1	88.8	79.2	84.6	79.7	76.0	70.6	57.4	75.0	73.5
DFN (I3D [6])	85.7	84.0	77.9	81.2	86.9	92.3	91.7	81.4	90.1	81.5	78.4	**74.9**	**63.5**	76.8	**75.0**
ASFormer [67](Br-prompt [36])	89.2	87.8	81.3	83.8	88.1	94.1	92.0	83.0	91.6	81.2			-		
DFN (Br-prompt [36])	**89.4**	**88.4**	**82.4**	**83.9**	**88.2**	**96.0**	**93.9**	**88.1**	**94.0**	**84.2**			-		

Table 2: Ablation study of self-attention (SA) and Fourier token mixer (FTM) based on ASFormer [67] baseline on the GTEA datasets. 'Order' indicates the order of feature aggregation in FTM; 't', 's', 'm' indicate temporal feature extractor, spectral feature extractor and multi-layer perceptron, respectively, '+' means the concatenation in parallel, while '.' means the sequential aggregation; 'SA$^-$' indicates the ASFormer with self-attention with half the number of layers and doubled the base ($2^k \rightarrow 4^k$), which is the same as FTM-based networks.

	Param.	Orders	F1@{10,25,50}			Edit	Acc
SA	1.13M	t.t.m	94.1	92.0	83.0	91.6	81.2
SA$^-$	0.55M	t.t.m	92.7	91.8	82.5	90.4	81.1
FTM	0.48M	m.t.s	93.5	92.3	85.6	91.0	82.8
	0.48M	s.t.m	91.7	89.5	79.4	88.6	81.2
	0.49M	t+s.m	92.8	91.6	81.4	90.9	83.0
	0.48M	t.s.m	**94.5**	**93.1**	**86.8**	**93.4**	**83.5**

mechanism in ASFormer [67] on Br-prompted [36] visual features. As shown in Table 2, the Fourier Network (FN) (*i.e.*, with Fourier token mixer) with fewer parameters and lower computational cost achieves better performance than the one with self-attention mechanism, indicating that it is better suited for the task of temporal action segmentation. It is worth noting that compared with the self-attention baseline (ASFormer [67]), FN still has a large improvement with half the number of layers by doubled the atrous rates, which proves the effectiveness of spectral calculations. Furthermore, we

Table 3: Ablation study results of the number of blocks with Fourier token mixer based on ASFormer [67].

(#) blocks	Param.	GFLOPS	F1@{10,25,50}			Edit	Acc
2	0.26M	0.31	89.3	87.5	79.2	83.1	80.0
5	0.48M	0.58	94.5	93.1	86.8	93.4	83.5
8	0.71M	0.85	95.8	93.4	87.0	93.8	83.9
10	0.89M	1.04	96.0	93.5	87.2	93.8	83.7

investigate the optimal mixing order of spectral features with temporal features in our proposed network, as shown in Table 2. The conclusion is that mixing spectral features with temporal features in the sequential order can further improve performance in temporal segmentation tasks. In addition, the experimental results show that the inductive bias introduced in Temporal Feature Extractors is important for achieving good performance. It is reasonable because inductive bias allows the model to generalize better to unseen data, by making certain assumptions about the structure of the data based on prior knowledge or experience. As a result, we propose to construct each neural block in a 'temporal → spectral → mlp' order, and add an extra residual connection on the spectral feature extractor, consist to the architecture shown in Figure 4.

Number of blocks in FN:. To evaluate the impact of size of model with Fourier token mixer, we compare different number of blocks. As shown in Table 3, when the number of blocks increases, the amount of parameters and the computational costs also increase accordingly, as well as the performance. This is because the stacking of blocks brings a larger receptive field and better non-locality. However, considering the law of diminishing marginal utility and

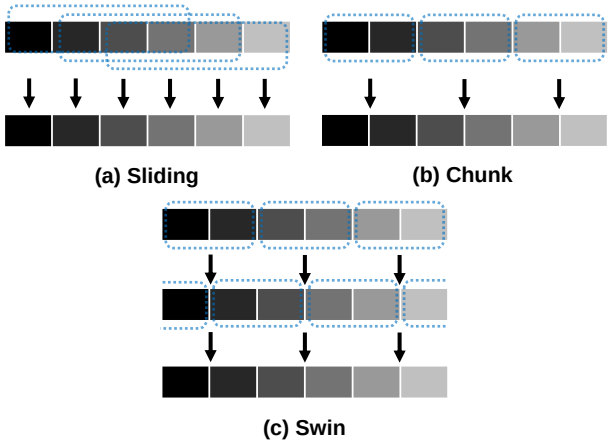

Figure 5: Comparison of different temporal window strategies. (a): Sliding strategy, which distinguishes the local information of each token. But the calculation of each token is generally independent, which thus requires high computational cost. (b): Chunk strategy that divides the sequence into chunks. Each of the chunk is computed independently. Although this strategy is computationally fast, the tokens between chunks lack information interaction. (c): Swin strategy, which adopts a shift operation when dividing chunks, to ensure the information exchange between adjacent chunks without any extra computational cost.

Table 4: Ablation study results of different window strategies on GTEA datasets.

Strategies	GFLOPS	F1@{10,25,50}			Edit	Acc
Sliding	9.0	95.1	93.5	86.9	93.6	84.0
Chunk	0.5	93.2	92.9	82.8	91.5	83.3
Swin	0.5	94.5	93.1	86.8	93.4	83.5

the accuracy-complexity trade-off of the model, we finally choose the number of blocks to be 5.

Choice of temporal windows: To compare different strategies of generating temporal windows in the Fourier Network, we conduct ablation studies with the same window sizes. The results are shown in Table 4. Generally speaking, chunk or swin strategy can save about $O(k)$ computational complexity when compared with the sliding strategy, where k is the window size. In addition, the sliding strategy also consumes a lot of GPU memory during the back-propagation calculation. Therefore, ASFormer [67] uses different masks for tokens to implement sliding strategy. This is because matrix products satisfies the distributive property. Unfortunately, this property is not satisfied in the Fourier transform. Sliding Discrete Fourier Transform (SDFT) [28] proves that, given the sequence $\mathbf{x}$, when the window of size m is shifted from $n-1$ to n, the DFT sequence $X(n)$ is totally changed, which is formulated as: $X_k(n) = [X_k(n-1) - x(n-m) + x(n)] \cdot e^{j2\pi k/m}$. Furthermore, this is harder to simplify after the non-linear operations (*e.g.*, ModReLU). Therefore, we adopt the swin strategy similar to that in SwinTransformer [40], which achieves non-trivial improvement in Table 4.

Table 5: Ablation study results of diffusion model and Frame-Wise Classification (FWC.) baseline model on GTEA dataset. We adopt the network with Fourier Token Mixers as baseline model. 'Steps' indicates the number of refining/denoising steps for segmentation; 's^*' means all parameters are shared weights on Refinement Stages; '$\mathcal{L}^c$' indicates additional use of classification losses for better visual features, see more explanation in main text.

#	Method	Steps	Param.	F1@{10,25,50}			Edit	Acc
1	FWC.	3	0.48M	94.5	93.1	86.8	93.4	83.5
2	FWC.	0	0.16M	60.8	59.2	51.6	50.4	76.9
3	FWC.s^*	3	0.27M	93.5	92.4	85.7	91.4	82.7
4	FWC.	10	1.26M	93.5	92.1	85.2	90.1	80.8
5	FWC.s^*	10	0.27M	93.8	92.7	86.5	91.8	81.2
6	Diff.	100	0.49M	95.8	93.6	87.8	93.7	84.0
7	Diff.$\mathcal{L}^c$	100	0.51M	**96.0**	**93.9**	**88.1**	**94.0**	**84.2**

Table 6: Ablation study results of the number of stages on GTEA datasets. 'U-C' indicates 'UnConditional', and 'C' indicates 'Conditional'. The total steps of diffusion process $M = 100$, and the number of unconditional stages is always set to 1, while the number of steps at each 'C' stage is the same.

number of U-C steps	number of C stages	F1@{10,25,50}			Edit	Acc
0	1	95.3	93.3	87.5	93.4	83.5
0	2	95.8	93.7	87.8	93.8	84.1
0	3	95.5	93.6	87.6	93.6	83.8
10	2	95.9	93.7	87.5	93.6	84.2
20	2	**96.0**	**93.9**	**88.1**	**94.0**	**84.2**
30	2	95.6	93.6	87.5	93.5	83.7
50	2	95.0	93.0	87.2	93.1	83.4

Table 7: Ablation study results of total diffusion steps M on GTEA datasets. The number of steps in each stage maintains the same proportion.

M	GFLOPS	F1@{10,25,50}			Edit	Acc
5	0.5	94.8	93.2	86.9	93.6	83.5
10	0.6	95.1	93.6	87.2	93.7	84.0
100	0.8	96.0	93.9	88.1	94.0	84.2
1000	3.5	95.9	93.9	88.3	94.0	84.2

Influence of steps of diffusion / denoising process: To evaluate the influence of the diffusion steps on the performance of the model, we conduct ablation studies by varying M from 5 to 1000. The results are shown in Table 7. Due to the recurrent calculation of f_θ, the

Table 8: Comparison of the number of parameters and FLOPs of our proposed method with other methods on GTEA datasets using I3D [6] features. The FLOPs are calculated using an input with the fixed length of 1000 frames.

Method	Param.	GFLOPS	F1@{10,25,50}			Edit	Acc
MS-TCN+ [37]	1.0M	1.0	88.8	85.7	76.0	83.5	80.1
ASRF [27]	1.3M	1.3	89.4	87.8	79.8	83.7	77.3
HASR [1]	18.8M	12.8	89.2	87.2	74.8	84.5	76.9
ETSN [38]	0.8M	**0.8**	91.1	90.0	77.9	86.2	78.2
DPRN [45]	4.1M	7.0	**92.9**	92.0	82.9	90.9	**82.0**
ASFormer [67]	1.1M	1.7	90.1	88.8	79.2	84.6	79.7
MSD + ASFormer [67]	1.3M	2.0	91.8	91.5	81.2	89.7	80.9
DFN	**0.5M**	**0.8**	92.3	91.7	81.4	90.1	81.5

computational costs grow with the increase of the total number of steps. Considering the trade-off between the model complexity and performance, we empirically set M to 100.

Effectiveness of diffusion models: To evaluate whether the proposed diffusion model can handle over-segmentation, we make ablation studies on baseline network using frame-wise classification or diffusion. The frame-wise classification based methods always adopt some Refinement Stages [15, 37] (similarly, Decoders [67], Temporal Reconstruction Networks (TRN) [45]) to further refine the segmentation results to alleviate over-segmentation. However, this part of the network usually requires a large number of parameters and computations to achieve the goal (about 2/3 of total network), as shown in Table 5.2. Moreover, due to the uncontrollable input distribution and the indistinguishable goal of stages, sharing parameters or stacking structure effects little (in Table 5, #3-#5). As for the diffusion models, the difference is that each stage is pre-defined as a step i during denoising process, controlled by β_i and the diffusion time embeddings. As each step in the diffusion process is defined as adding Gaussian noise, the entire diffusion model is usually designed to be light-weighted, shared and recurrently used to save parameters and GFLOPS. Specifically, we adopt a simple MLP with some temporal convolutional layers conditioned on visual features from the network with Fourier token mixers. As shown in Table 5 #6, the diffusion model achieves better $F1$ and *Edit* scores, implying better performance in solving over-segmentation. In Table 5 #7, we further improve performance by using additional supervisions on visual features (cross-entropy loss and T-MSE loss [37]). Additionally, we also visualize the output of diffusion model and compare it with ground truth annotations. As we can see, the diffusion model can effectively alleviate the over-segmentation problems. The visualization results are shown in Figure 6.

Multi-stage diffusion: To evaluate the effectiveness of the proposed multi-stage diffusion model, we vary it with a single-stage diffusion model. For the single-stage diffusion model, we only use the conditional stage, which takes the visual features during denoising process. The results are shown in Table 6. As we can see, the multi-stage diffusion model achieves better results than the single-stage diffusion model on both datasets, indicating that the

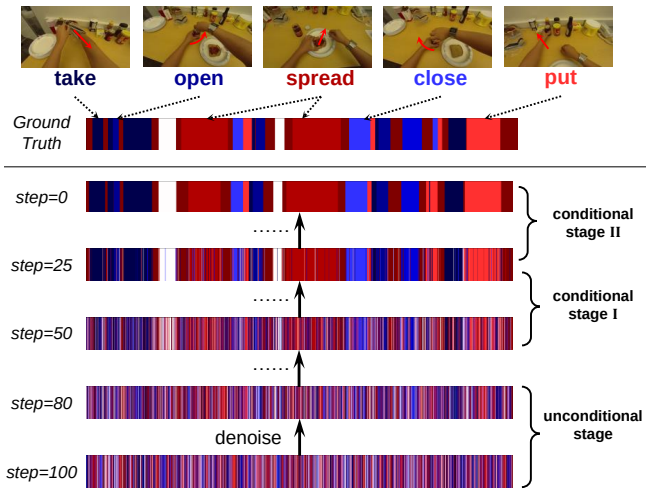

Figure 6: Visualization of the diffused output at each stage and ground truth annotations (GT). The example is 'S1_Pealate_C1' from the GTEA dataset.

unconditional denoising process in the first stage helps to learn a better prior distribution and improves the performance of the final segmentation.

Parameter and complexity analyse: To evaluate the effectiveness of the proposed model in reducing the amount of parameters and calculations, we compare the number of parameters and FLOPs of our model with that of state-of-the-art methods, as shown in Table 8. Compared to ETSN [38], DFN outperforms on all the metrics, indicating its effectiveness; compared to DPRN [45], DFN only uses about 12% number of parameters and 11% computational complexity to achieves competitive performance (about 0.3% $\sim$ 1.5% drops). It is notable that multi-stage diffusion models (MSD) can be transferred to most temporal segmentation networks with a small cost of parameters and computation, as shown in the Table 8, MSD+ASFormer [67]. The result shows that this can further improve the performance.

5 CONCLUSION

We present Diffused Fourier network, an efficient solution for human action segmentation via multi-stage diffusion models and Fourier Token Mixer. DFN utilizes the strength of diffusion models to solve over-segmentation, and combines with Fourier Token Mixer to efficiently capture both local and global information. Through extensive experiments results, we demonstrate that DFN outperforms existing methods while using fewer parameters and computations. Furthermore, we conduct ablation studies to analyze the contributions of each component in DFN and provide insights into future research directions. Our work shows the potential of combining diffusion models and token mixing mechanisms for efficient and accurate temporal action segmentation tasks.

ACKNOWLEDGMENTS

This work is supported by National Key R&D Program of China (2022ZD0160305) and Beijing Natural Science Foundation (Z190001).

REFERENCES

[1] Hyemin Ahn and Dongheui Lee. 2021. Refining Action Segmentation with Hierarchical Video Representations. In *2021 IEEE/CVF International Conference on Computer Vision*. IEEE, 16282–16290.

[2] Tomer Amit, Eliya Nachmani, Tal Shaharabany, and Lior Wolf. 2021. SegDiff: Image Segmentation with Diffusion Probabilistic Models. *CoRR* abs/2112.00390 (2021).

[3] Martín Arjovsky, Amar Shah, and Yoshua Bengio. 2016. Unitary Evolution Recurrent Neural Networks. In *Proceedings of the 33nd International Conference on Machine Learning (JMLR Workshop and Conference Proceedings, Vol. 48)*, Maria-Florina Balcan and Kilian Q. Weinberger (Eds.). JMLR.org, 1120–1128.

[4] Nadine Behrmann, S. Alireza Golestaneh, Zico Kolter, Jürgen Gall, and Mehdi Noroozi. 2022. Unified Fully and Timestamp Supervised Temporal Action Segmentation via Sequence to Sequence Translation. In *European Conference on Computer Vision (Lecture Notes in Computer Science, Vol. 13695)*, Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (Eds.). Springer, 52–68.

[5] Matthew Brand and Vera M. Kettnaker. 2000. Discovery and Segmentation of Activities in Video. *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 8 (2000), 844–851.

[6] João Carreira and Andrew Zisserman. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 4724–4733.

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. 2018. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 4 (2018), 834–848.

[8] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. 2022. DiffusionDet: Diffusion Model for Object Detection. *CoRR* abs/2211.09788 (2022).

[9] Ting Chen, Lala Li, Saurabh Saxena, Geoffrey E. Hinton, and David J. Fleet. 2022. A Generalist Framework for Panoptic Segmentation of Images and Videos. *CoRR* abs/2210.06366 (2022).

[10] Lu Chi, Borui Jiang, and Yadong Mu. 2020. Fast Fourier Convolution. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.), Vol. 33. 4479–4488.

[11] Robert T. Collins, Christophe Biernacki, Gilles Celeux, Alan J. Lipton, Gérard Govaert, and Takeo Kanade. 2000. Introduction to the Special Section on Video Surveillance. *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 8 (2000), 745–746.

[12] Trevor Darrell and Alex Pentland. 1993. Space-time gestures. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 335–340.

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *9th International Conference on Learning Representations*. OpenReview.net.

[14] Dazhao Du, Bing Su, Yu Li, Zhongang Qi, Lingyu Si, and Ying Shan. 2022. Do we really need temporal convolutions in action segmentation. *arXiv preprint arXiv:2205.13425* 2 (2022), 13.

[15] Yazan Abu Farha and Jürgen Gall. 2019. MS-TCN: Multi-Stage Temporal Convolutional Network for Action Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 3575–3584.

[16] Alireza Fathi, Xiaofeng Ren, and James M. Rehg. 2011. Learning to recognize objects in egocentric activities. In *IEEE Conference on Computer Vision and Pattern Recognition*. 3281–3288.

[17] Shang-Hua Gao, Qi Han, Zhong-Yu Li, Pai Peng, Liang Wang, and Ming-Ming Cheng. 2021. Global2Local: Efficient Structure Search for Video Action Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 16805–16814.

[18] Pallabi Ghosh, Yi Yao, Larry S. Davis, and Ajay Divakaran. 2020. Stacked Spatio-Temporal Graph Convolutional Networks for Action Segmentation. In *IEEE Winter Conference on Applications of Computer Vision*. IEEE, 565–574.

[19] Zhangxuan Gu, Haoxing Chen, Zhuoer Xu, Jun Lan, Changhua Meng, and Weiqiang Wang. 2022. DiffusionInst: Diffusion Model for Instance Segmentation. *CoRR* abs/2212.02773 (2022).

[20] Mahmudul Hasan and Amit K. Roy-Chowdhury. 2015. A Continuous Learning Framework for Activity Recognition Using Deep Hybrid Feature Models. *IEEE Trans. Multim.* 17, 11 (2015), 1909–1922.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[22] Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415* (2016).

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.).

[24] Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. 2022. Cascaded Diffusion Models for High Fidelity Image Generation. *J. Mach. Learn. Res.* 23 (2022), 47:1–47:33.

[25] Jonathan Ho, Tim Salimans, Alexey A. Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. 2022. Video Diffusion Models. *CoRR* abs/2204.03458 (2022).

[26] Yifei Huang, Yusuke Sugano, and Yoichi Sato. 2020. Improving Action Segmentation via Graph-Based Temporal Reasoning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Computer Vision Foundation / IEEE, 14021–14031.

[27] Yuchi Ishikawa, Seito Kasai, Yoshimitsu Aoki, and Hirokatsu Kataoka. 2021. Alleviating Over-segmentation Errors by Detecting Action Boundaries. In *IEEE Winter Conference on Applications of Computer Vision*. 2321–2330.

[28] Eric Jacobsen and Richard G. Lyons. 2003. The sliding DFT. *IEEE Signal Process. Mag.* 20, 2 (2003), 74–80.

[29] Yitzhak Katznelson. 2004. *An introduction to harmonic analysis*. Cambridge University Press.

[30] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations*, Yoshua Bengio and Yann LeCun (Eds.).

[31] Hilde Kuehne, Ali Bilgin Arslan, and Thomas Serre. 2014. The Language of Actions: Recovering the Syntax and Semantics of Goal-Directed Human Activities. In *IEEE Conference on Computer Vision and Pattern Recognition*. 780–787.

[32] Colin Lea, Michael D. Flynn, René Vidal, Austin Reiter, and Gregory D. Hager. 2017. Temporal Convolutional Networks for Action Segmentation and Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1003–1012.

[33] Colin Lea, Austin Reiter, René Vidal, and Gregory D. Hager. 2016. Segmental Spatiotemporal CNNs for Fine-Grained Action Segmentation. In *European Conference on Computer Vision*, Vol. 9907. 36–52.

[34] James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontañón. 2022. FNet: Mixing Tokens with Fourier Transforms. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz (Eds.). Association for Computational Linguistics, 4296–4313.

[35] Peng Lei and Sinisa Todorovic. 2018. Temporal Deformable Residual Networks for Action Segmentation in Videos. In *IEEE Conference on Computer Vision and Pattern Recognition*. 6742–6751.

[36] Muheng Li, Lei Chen, Yueqi Duan, Zhilan Hu, Jianjiang Feng, Jie Zhou, and Jiwen Lu. 2022. Bridge-Prompt: Towards Ordinal Action Understanding in Instructional Videos. *CoRR* abs/2203.14104 (2022).

[37] Shi-Jie Li, Yazan AbuFarha, Yun Liu, Ming-Ming Cheng, and Juergen Gall. 2020. MS-TCN++: Multi-Stage Temporal Convolutional Network for Action Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* (2020). early access. doi: 10.1109/TPAMI.2020.3021756.

[38] Yunheng Li, Zhuben Dong, Kaiyuan Liu, Lin Feng, Lianyu Hu, Jie Zhu, Li Xu, Yuhan wang, and Shenglan Liu. 2021. Efficient Two-Step Networks for Temporal Action Segmentation. *Neurocomputing* 454 (2021), 373–381.

[39] Yanyu Li, Geng Yuan, Yang Wen, Ju Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. 2022. Efficientformer: Vision transformers at mobilenet speed. *Advances in Neural Information Processing Systems* 35 (2022), 12934–12949.

[40] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *2021 IEEE/CVF International Conference on Computer Vision*. IEEE, 9992–10002.

[41] Fengjun Lv and Ramakant Nevatia. 2006. Recognition and Segmentation of 3-D Human Action Using HMM and Multi-class AdaBoost. In *European Conference on Computer Vision (Lecture Notes in Computer Science, Vol. 3954)*, Ales Leonardis, Horst Bischof, and Axel Pinz (Eds.). Springer, 359–372.

[42] D. Marr and Lucia Vaina. 1982. Representation and Recognition of the Movements of Shapes. *Proceedings of the Royal Society of London. Series B, Biological Sciences* 214, 1197 (1982), 501–524.

[43] Tomás Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association*, Takao Kobayashi, Keikichi Hirose, and Satoshi Nakamura (Eds.). ISCA, 1045–1048.

[44] Louis-Philippe Morency, Ariadna Quattoni, and Trevor Darrell. 2007. Latent-Dynamic Discriminative Models for Continuous Gesture Recognition. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA*. IEEE Computer Society.

[45] Junyong Park, Daekyum Kim, Sejoon Huh, and Sungho Jo. 2022. Maximization and restoration: Action segmentation through dilation passing and temporal reconstruction. *Pattern Recognit.* 129 (2022), 108764.

[46] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. (2017).

[47] Siyuan Qi, Baoxiong Jia, and Song-Chun Zhu. 2018. Generalized Earley Parser: Bridging Symbolic Grammars and Sequence Data for Future Prediction. In *Proceedings of the 35th International Conference on Machine Learning*, Jennifer G. Dy and Andreas Krause (Eds.). 4168–4176.

[48] Lawrence R. Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77, 2 (1989), 257–286.

[49] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.

[50] Yong Rui and P. Anandan. 2000. Segmenting Visual Actions Based on Spatio-Temporal Motion Patterns. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 1111–1118.

[51] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *CoRR* abs/2205.11487 (2022).

[52] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. 2023. Image Super-Resolution via Iterative Refinement. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 4 (2023), 4713–4726.

[53] Hasim Sak, Andrew W. Senior, and Françoise Beaufays. 2014. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association*, Haizhou Li, Helen M. Meng, Bin Ma, Engsiong Chng, and Lei Xie (Eds.). ISCA, 338–342.

[54] Dipika Singhania, Rahul Rahaman, and Angela Yao. 2021. Coarse to Fine Multi-Resolution Temporal Convolutional Network. *CoRR* abs/2105.10859 (2021).

[55] Cristian Sminchisescu, Atul Kanaujia, and Dimitris N. Metaxas. 2006. Conditional models for contextual human motion recognition. *Comput. Vis. Image Underst.* 104, 2-3 (2006), 210–220.

[56] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. Denoising Diffusion Implicit Models. In *9th International Conference on Learning Representations*. OpenReview.net.

[57] Yang Song and Stefano Ermon. 2019. Generative Modeling by Estimating Gradients of the Data Distribution. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 11895–11907.

[58] Sebastian Stein and Stephen J. McKenna. 2013. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 729–738.

[59] Charles Sutton and Andrew McCallum. 2012. An Introduction to Conditional Random Fields. *Found. Trends Mach. Learn.* 4, 4 (2012), 267–373.

[60] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. 2016. Instance Normalization: The Missing Ingredient for Fast Stylization. *CoRR* abs/1607.08022 (2016). arXiv:1607.08022 http://arxiv.org/abs/1607.08022

[61] Nam N. Vo and Aaron F. Bobick. 2014. From Stochastic Grammar to Bayes Network: Probabilistic Parsing of Complex Activity. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society.

[62] Jiahui Wang, Zhenyou Wang, Shanna Zhuang, and Hui Wang. 2022. Cross-Enhancement Transformer for Action Segmentation. *CoRR* abs/2205.09445 (2022).

[63] Zhenzhi Wang, Ziteng Gao, Limin Wang, Zhifeng Li, and Gangshan Wu. 2020. Boundary-Aware Cascade Networks for Temporal Action Segmentation. In *European Conference on Computer Vision*, Vol. 12370. 34–51.

[64] Daniel Weinland, Rémi Ronfard, and Edmond Boyer. 2011. A survey of vision-based methods for action representation, segmentation and recognition. *Comput. Vis. Image Underst.* 115, 2 (2011), 224–241.

[65] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2022. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. *CoRR* abs/2210.02186 (2022).

[66] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021*, Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (Eds.). 22419–22430.

[67] Fangqiu Yi, Hongyu Wen, and Tingting Jiang. 2021. ASFormer: Transformer for Action Segmentation. *CoRR* abs/2110.08568 (2021).

[68] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. 2022. MetaFormer is Actually What You Need for Vision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 10809–10819.

[69] Lihi Zelnik-Manor and Michal Irani. 2001. Event-Based Analysis of Video. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 123–130.

[70] Junbin Zhang, Pei-Hsuan Tsai, and Meng-Hsun Tsai. 2022. Semantic2Graph: Graph-based Multi-modal Feature for Action Segmentation in Videos. *CoRR* abs/2209.05653 (2022).

[71] Roland S. Zimmermann, Lukas Schott, Yang Song, Benjamin A. Dunn, and David A. Klindt. 2021. Score-Based Generative Classifiers. *CoRR* abs/2110.00473 (2021).