FIFA: Fast Inference Approximation for Action Segmentation

Yaser Souri[1], Yazan Abu Farha[1], Fabien Despinoy[2], Gianpiero Francesca[2], Juergen Gall[1]

[1]University of Bonn, [2]Toyota Motor Europe

{lastname}@iai.uni-bonn.de, {firsname.lastname}@toyota-motor.com

Abstract

We introduce FIFA, a fast approximate inference method for action segmentation and alignment. Unlike previous approaches, FIFA does not rely on an expensive Viterbi decoding for optimization. Instead, it uses an approximate differentiable energy function that can be minimized using gradient-descent. FIFA is a general approach that can replace exact inference improving its speed by more than 5 times while maintining its peformance. FIFA also provides a better speed vs. accuracy tradeoff compared to exact inference. We apply FIFA on top of state-of-the-art approaches for weakly supervised action segmentation and alignment as well as fully supervised action segmentation. FIFA achives state-of-the-art results on most metrics on the three tasks.

1. Introduction

Video action segmentation is a promising area of research where deep learning methodologies allow nowadays us to achieve unprecedented results. With the success of fully supervised approaches [22, 1, 36] that predict the frame-wise action labels [1, 27], researchers recently paid more attention to weakly supervised approaches and especially the ones trained using transcripts of actions [31, 26, 33]. However, their main weakness is still the computationally expensive Viterbi decoding during inference time [19, 30, 31, 26, 33]. Indeed, these approaches combine an action length model with the predicted frames-wise probabilities to predict a sequence of frame-wise action labels that minimizes an energy function. Despite the achieved improvements on the accuracy using such decoding step, it makes the inference procedure extremely slow.

In this paper, we propose FIFA, a fast approximate inference procedure that achieves comparable performance with respect to the Viterbi decoding inference at a fraction of the computational time. Instead of relying on dynamic programming, we formulate the energy function as an approximate differentiable function of learned segment lengths parameters and use gradient-descent-based methods to search for a configuration that minimizes the approximate energy function. Given a transcript of actions and the corresponding initial lengths configuration, we define the energy function as a sum over segment level energies. The segment level energy consists of two terms: a length energy term that penalizes the deviations from a global length model and an observation energy term that measures the compatibility between the current configuration and the predicted frame-wise probabilities. A naive approach to model the observation energy would be, to sum up the negative log probabilities of the action labels that are defined based on the length configuration. Nevertheless, such an approach is not differentiable with respect to the segment lengths. In order to optimize the energy using gradient descent-based methods, the observation energy has to be differentiable with respect to the segment lengths. To this end, we construct a plateau-shaped mask for each segment which temporally locates the segment within the video. This mask is parameterized by the segment lengths, the position in the video, and a sharpness parameter. The observation energy is then defined as a product of a segment mask and the predicted frame-wise negative log probabilities, followed by a sum pooling operation. Finally, a gradient descent-based method is used to find a configuration for the segment lengths that minimizes the total energy.

FIFA is a general inference approach and can be applied at test time on top of other action segmentation approaches for fast inference. We evaluate our approach on top of the state-of-the-art methods for weakly supervised temporal action segmentation, weakly supervised action alignment, and fully supervised action segmentation. Results on the Breakfast dataset [17] show that FIFA achieves state-of-the-art results on most metrics on the three tasks. Compared to the exact inference using the Viterbi decoding, FIFA is at least 5 times faster. Furthermore, FIFA provides a better speed vs. accuracy tradeoff compared to exact inference.

2. Related Work

In this section we highlight relevant works addressing fully and weakly supervised action segmentation that have been recently achieved.

Fully Supervised Action Segmentation. In fully supervised action segmentation, frame-level labels are used for training. Initial attempts for action segmentation applied action classifiers on a sliding window over the video frames [32, 16]. However, these approaches did not capture the dependencies between the action segments. With the objective of capturing the context over long video sequences, context free grammars [35, 29] or hidden Markov models (HMMs) [23, 18, 20] are typically combined with framewise classifiers. Recently, temporal convolutional networks showed good performance for the temporal action segmentation task using encoder-decoder architectures [22, 25] or even multi-stage architectures [1, 27]. Many approaches further improve the multi-stage architectures by applying post-processing based on boundary-aware pooling operation [36, 14] or graph-based reasoning [13]. Nonetheless, despite the success of fully supervised approaches, the high cost of frame-level annotations hinders their applicability on large scale.

Weakly Supervised Action Segmentation. To reduce the annotation cost, many approaches that rely on a weaker form of supervision have been proposed. Earlier approaches apply discriminative clustering to align video frames to movie scripts [7]. Bojanowski *et al.* [4] proposed to use as supervision the transcripts in the form of ordered lists of actions. Indeed, many approaches rely on this form of supervision to train a segmentation model using connectionist temporal classification [12], dynamic time warping [5] or energy-based learning [26]. In [6], an iterative training procedure is used to refine the transcript. A soft labeling mechanism is further applied at the boundaries between action segments. Kuehne *et al.* [19] applied a speech recognition system based on a HMM and Gaussian mixture model (GMM) to align video frames to transcripts. The approach generates pseudo ground truth labels for the training videos and iteratively refine them. A similar idea has been recently used in [30, 20]. Richard *et al.* [31] combined the framewise loss function with the Viterbi algorithm to generate the target labels. At inference time, these approaches iterate over the training transcripts and select the one that matches best the testing video. By contrast, Souri *et al.* [33] predict the transcript besides the frame-wise scores at inference time.

Energy-Based Inference. In energy-based inference methods, gradient descent is used at inference time as described in [24]. The goal is to minimize an energy function that measures the compatibility between the input variables and the predicted variables. This idea has been exploited for many structured prediction tasks such as image generation [8, 15], machine translation [11] and structured prediction energy networks [3]. Belanger and McCallum [2] relaxed the discrete output space for multi-label classification tasks to a continuous space and used gradient descent to approximate the solution. Gradient-based methods have also been used for other applications such as generating adversarial examples [10] and learning text embeddings [21].

3. Background

The following sections introduce all the concepts and notations required to understand the proposed FIFA methodology.

3.1. Action Segmentation

In action segmentation, we want to temporally localize all the action segments occurring in a video. In this paper, we consider the case where the actions are from a predefined set of M classes (a background class is used to cover uninteresting parts of a video). The input video of length T is usually represented as a set of d dimensional features vectors $x_{1:T} = (x_1, \dots, x_T)$. These features are extracted offline and are assumed to be the input to the action segmentation model. The output of action segmentation can be represented in two ways:

- Frame-wise representation $y_{1:T} = (y_1, \dots, y_T)$ where y_t represents the action label at time t.
- Segment-wise representation $s_{1:N} = (s_1, \dots, s_N)$ where segment s_n is represented by both the action label of the segment c_n and its corresponding length ℓ_n i.e. $s_n = (c_n, \ell_n)$. The ordered list of actions $c_{1:N}$ is usually referred to as the *transcript*.

These two representations are equal and redundant i.e. it is possible to compute one from the other. In order to transfer from the segment-wise to the frame-wise representation, we introduce a function $\alpha(t; c_{1:N}, \ell_{1:N})$ which outputs the action label at frame t given the segment-wise labeling.

The target labels to train a segmentation model, depend on the level of supervision. In fully supervised action segmentation [1, 27, 36], the target label for each frame is basically provided. However, in weakly supervised approaches [31, 26, 33] only the ordered list of action labels are provided during training while their lengths are unknown.

Recent fully supervised approaches for action segmentation like MSTCN [1] and its variants directly predict the frame-wise representation $y_{1:T}$. During testing, the prediction is made by choosing the action label with the highest probability for each frame.

Conversely, recent weakly supervised action segmentation approaches like NNV [31] and follow-up work includes an inference stage during testing where they explicitly predict the segment-wise representation. This inference stage involves a dynamic programming algorithm for solving an optimization problem which is a computational bottleneck for these approaches.

3.2. Inference in Action Segmentation

During testing, the inference stage involves an optimization problem to find the most likely segmentation for the input video i.e.,

$$c_{1:N}, \ell_{1:N} = \underset{\hat{c}_{1:N}, \hat{\ell}_{1:N}}{\operatorname{argmax}} \Big\{ p(\hat{c}_{1:N}, \hat{\ell}_{1:N} | x_{1:T}) \Big\}. \tag{1}$$

Given the transcript $c_{1:N}$, the inference stage boils down to find the segment lengths $\ell_{1:N}$ by aligning the transcript to the input video i.e.,

$$\ell_{1:N} = \underset{\hat{\ell}_{1:N}}{\operatorname{argmax}} \Big\{ p(\hat{\ell}_{1:N} | x_{1:T}, \hat{c}_{1:N}) \Big\}. \tag{2}$$

In approaches like NNV [31] and CDFL [26], the transcript is found by iterating over the transcripts seen during training and selecting the transcript that achieves the most likely alignment by optimizing (2). In MuCon [33], the transcript is predicted by a sequence to sequence network.

The probability defined in (2) is broken down by making independences assumption between frames

$$\begin{aligned} p(\hat{\ell}_{1:N} | x_{1:T}, \hat{c}_{1:N}) = & \prod_{t=1}^{T} p\big(\alpha(t; c_{1:N}, \ell_{1:N}) | x_t\big) \\ & \cdot \prod_{n=1}^{N} p\big(\ell_n | c_n\big) \end{aligned} \tag{3}$$

where $p\big(\alpha(t)|x_t\big)$ is referred to as the observation model and $p\big(\ell_n|c_n\big)$ as the length model. The observation model estimates the frame-wise action probabilities and is implemented using a neural network. The length model is used to constrain the inference defined in (2) with the assumption that the length of segments for the same action follow a particular probability distribution. The segment length is usually modelled by a Poisson distribution with a class dependent mean parameter λ_n i.e.,

$$p\big(\ell_n | c_n\big) = \frac{\lambda_n^{\ell_n} \exp(-\lambda_n)}{\ell_n!}. \tag{4}$$

3.3. Exact Inference

The NNV approach [31] proposes an exact solution to the inference problem (2) using a Viterbi-like dynamic programming method. This dynamic programming approach was later adopted by CDFL [26] and MuCon [33]. First an auxiliary function $Q(t, \ell, n)$ is defined that yields the best probability score for a segmentation up to frame t satisfying the following conditions:

- the length of the last segment is ℓ,
- the last segment was the nth segment with label c_n.

The function Q can be computed recursively. The following two cases are distinguished. The first case defines when no new segment is hypothesized, i.e $\ell > 1$. Then,

$$Q(t, \ell, n) = Q(t-1, \ell-1, n) \cdot p(c_n | x_t), \tag{5}$$

with the current frame probability being multiplied with the value of the auxiliary function at the previous frame. The second case is a new segment being hypothesized at frame t, i.e. $\ell = 1$. Then,

$$\begin{aligned} & Q(t, \ell = 1, n) = \\ & \max_{\hat{\ell}} \Big\{ Q(t-1, \hat{\ell}, n-1) \cdot p(c_n | x_t) \cdot p(\hat{\ell} | c_{n-1})) \Big\}, \end{aligned} \tag{6}$$

where the optimization being calculated over all possible previous segments with length $\hat{\ell}$ and label c_{n-1}. Here the probability of the previous segment having length $\hat{\ell}$ and label c_{n-1} is being multiplied to the previous value of the auxiliary function.

The most likely alignment is given by

$$\underset{\ell}{max} \Big\{ Q(T, \ell, N) \cdot p(\ell | c_N) \Big\}. \tag{7}$$

The optimal lengths can be obtained by keeping track of the maximizing arguments $\hat{\ell}$ from (6).

3.4. Time Complexity of Exact Inference

The time complexity of the above exact inference is quadratic in the length of the video T and linear in the number of segments N. As input videos for action segmentation are usually long, it becomes computationally expensive to calculate. In practice, [31, 26, 33] limit the maximum size of each segment to a fixed value of $L = 2000$. The final time complexity of exact inference is $O(LNT)$. Furthermore, this optimization process is inherently not parallelizable. This is due to the $\max$ operation in (6). Experiments have shown [33, 34] that this inference stage is the main computational bottleneck of action segmentation approaches.

4. FIFA: Fast Inference Approximation

Our goal is to introduce a fast inference algorithm for action segmentation. We want the fast inference to be applicable in both weakly supervised and fully supervised action segmentation. We also want the fast inference to be

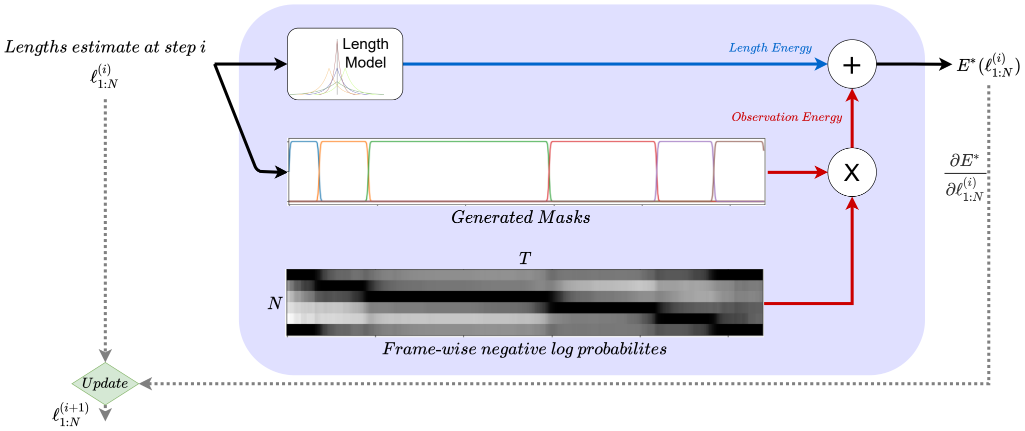

Figure 1. Overview of the FIFA optimization process. At each step in the optimization, using the current length estimates a set of masks are generated. Using the generated masks and the frame-wise negative log probabilities, the observation energy is calculated in an approximate but differentiable manner. The length energy is calculated from the current length estimate and added to the observation energy to calculate the total energy value. Taking the gradient of the total energy with respect to the length estimates we can update it using a stochastic gradient step.

flexible enough to work with different action segmentation methods. To this end, we introduce FIFA, a novel approach for fast inference for action segmentation.

In the following for brevity we write $\alpha(t; c_{1:N}, \ell_{1:N})$ simply as $\alpha(t)$. Maximizing probability (2) can be rewriten as minimizing the negative log of that probability

$$\max\Big\{p(\hat{\ell}_{1:N}|x_{1:T}, \hat{c}_{1:N})\Big\} = \\ \min\Big\{-\log\big(p(\hat{\ell}_{1:N}|x_{1:T}, \hat{c}_{1:N})\big)\Big\} \tag{8}$$

which we refer to as the energy $E(\ell_{1:N})$. Using (3) the energy can be rewritten as

$$\begin{aligned} E(\ell_{1:N}) &= -\log\Big(p(\ell_{1:N}|x_{1:T}, c_{1:N})\Big) \\ &= -\log\left(\prod_{t=1}^{T} p(\alpha(t)|x_t) \cdot \prod_{n=1}^{N} p\big(\ell_n|c_n\big)\right) \\ &= \underbrace{\sum_{t=1}^{T} -\log p\big(\alpha(t)|x_t\big)}_{E_o} + \underbrace{\sum_{n=1}^{N} -\log p\big(\ell_n|c_n\big)}_{E_\ell}. \end{aligned} \tag{9}$$

The first term in (9), E_o is referred to as the observation energy. This term calculates the cost of assigning the labels for each frame and will be calculated from the frame-wise probability estimates. The second term E_ℓ is referred to as the length energy. This term is the cost of each segment having a length given that we assume some average length for actions of a specific class.

We proposed to optimize the energy defined in (9) using SGD (Stochastic Gradient Descent) in order to avoid the need for time-consuming dynamic programming. We start with an initial estimate of the lengths (obtained from the length model of each approach or calculated from training data when available) and update our estimate to minimize the energy function.

As the energy function $E(\ell_{1:N})$ is not differentiable with respect to the lengths, we have to calculate a relaxed and approximate energy function $E^*(\ell_{1:N})$ that respects this mathematical property.

4.1. Approximate Differentiable Energy E^*

The energy function E as defined in (9) is not differentiable in two parts. First the observation energy term E_o is not differentiable because of the $\alpha(t)$ function. Second, the length energy term E_ℓ is not differentiable because it expects natural numbers as input and cannot be computed on real values which are dealt with in gradient-based optimization. Below we describe how we approximate and make each of the terms differentiable.

4.1.1 Approximate Differentiable Observation Energy

Consider a $N \times T$ matrix P containing negative log probabilities, i.e.

$$P[n, t] = -\log p(c_n|x_t). \tag{10}$$

Imagine a mask matrix M with the same size $N \times T$ where

$$M[n, t] = \begin{cases} 0 & if\ \alpha(t) \neq c_n \\ 1 & if\ \alpha(t) = c_n \end{cases}. \tag{11}$$

Using the mask matrix we can rewrite the observation energy term as

$$E_o = \sum_{t=1}^{T} \sum_{n=1}^{N} M[n,t] \cdot P[n,t]. \tag{12}$$

In order to make the observation energy term differentiable with respect to the length, we propose to construct an approximate differentiable mask matrix M^*. We use the following smooth and parametric plateau function

$$f(t|\lambda^c, \lambda^w, \lambda^s) = \frac{1}{(e^{\lambda^s(t-\lambda^c-\lambda^w)}+1)(e^{\lambda^s(-t+\lambda^c-\lambda^w)}+1)} \tag{13}$$

from [28]. This plateau function has three parameters and it is differentiable with respect to them: λ^c controls the center of the plateau, λ^w is the width and λ^s is the sharpness of the plateau function.

While the sharpness of the plateau functions λ^s used to construct the approximate mask M^* is fixed as a hyperparameter of our approach, the center λ^c and the width λ^w are computed from the lengths $\ell_{1:N}$. First we calculate the starting position of each plateau function b_n as

$$b_1 = 0, b_n = \sum_{n'=1}^{n-1} \ell_{n'}. \tag{14}$$

We can then define both the center and the width parameters of each plateau function as

$$\begin{aligned} \lambda_n^c &= b_n + \ell_n/2 \\ \lambda_n^w &= \ell_n/2 \end{aligned} \tag{15}$$

and define each row of the approximate mask as

$$M^*[n,t] = f(t|\lambda_n^c, \lambda_n^w, \lambda^s). \tag{16}$$

Now we can calculate a differentiable approximate observation energy similar to (12) as

$$E_o^* = \sum_{t=1}^{T} \sum_{n=1}^{N} M^*[n,t] \cdot P[n,t]. \tag{17}$$

4.1.2 Approximate Differentiable Length Energy

For the SGD optimization, we must relax the length values to be positive real values instead of natural numbers. As the Poisson distribution (4) is only defined on natural numbers, we propose to use a substitute distribution defined on real numbers. As a replacement, we experiment with a Laplace distribution and a Gaussian distribution. In both cases, the scale or the width parameter of the distribution is assumed to be fixed.

We can rewrite the length energy E_ℓ as the approximate length energy

$$E_\ell^*(\ell_{1:N}) = \sum_{n=1}^{N} -\log p(\ell_n|\lambda_{c_n}^\ell) \tag{18}$$

where $\lambda_{c_n}^\ell$ is the expected value for the length of a segment from the action c_n. In case of the Laplace distribution this length energy will be equal to

$$E_\ell^*(\ell_{1:N}) = \frac{1}{Z} \sum_{n=1}^{N} |\ell_n - \lambda_{c_n}^\ell|. \tag{19}$$

This means that the length energy will penalize any deviation from the expected average length linearly. Similarly, the following Gaussian distribution

$$E_\ell^*(\ell_{1:N}) = \frac{1}{Z} \sum_{n=1}^{N} |\ell_n - \lambda_{c_n}^\ell|^2 \tag{20}$$

means that the Gaussian length energy will penalize any deviation from the expected average length quadratically.

With the objective to maintain a positive value for the length during the optimization process, we estimate the length in log space and convert it to absolute space only in order to compute both the approximate mask matrix M^* and the approximate length energy E_ℓ^*.

4.1.3 Approximate Energy Optimization

The total approximate energy function is defined as a weighted sum of both the approximate observation and the approximate length energy functions

$$E^*(\ell_{1:N}) = E_o^*(\ell_{1:N}, Y) + \beta E_\ell^*(\ell_{1:N}) \tag{21}$$

where β is the multiplier for the length energy.

Given an initial length estimate $\ell_{1:N}^0$, we iteratively update this estimate to minimize the total energy. Figure 1 illustrates the optimization step for our approach. During each optimization step, we first calculate the energy E^* and then calculate the gradients of the energy with respect to the length values. Using the calculated gradients, we update the length estimate using a stochastic gradient descent update rule such as SGD or Adam. After a certain number of gradient steps we will finally predict the segment length.

During testing, if the transcript is provided then it is used (e.g. using the MuCon [33] approach or in a weakly supervised action alignment setting). However, if the latter is not known (e.g. in a fully supervised approach or CDFL [26] for weakly supervised action segmentation) we perform the optimization for each of the transcripts seen during training and select the most likely one.

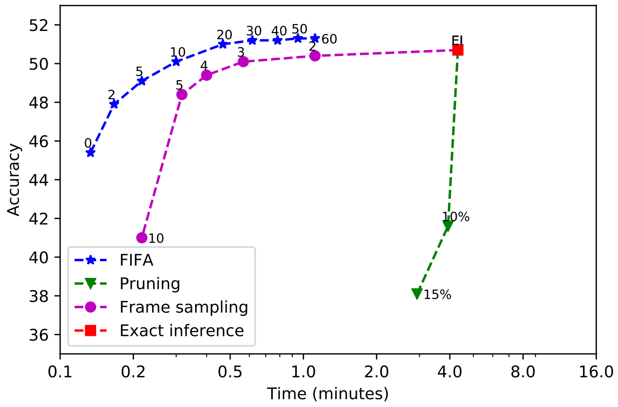

Figure 2. Speed vs. accuracy trade-off of different inference approaches applied to the MuCon method. Using FIFA we can achieve a better speed vs. accuracy trade-off compared to frame sampling or hypothesis pruning in exact inference.

The initial length estimates are calculated from the length model of each approach in case of weakly supervised processing whereas in fully supervised the average length of each action class is calculated from the training data and used as the initial length estimate. The initial length estimates are also used as the expected length parameters for the length energy calculations.

The optimization hyper-parameters like the choice of the optimizer, number of steps, and learning rate remain as the hyper-parameters of our approach.

4.1.4 Time Complexity

At each optimization step, the time complexity is $O(NT)$, where N is the number of segments and T is the length of the video because we must create the M^* matrix and calculate the element-wise multiplication. Overall, the FIFA time complexity is $O(MNT)$, where M is the number of optimization steps. Compared to the exact inference which has a time complexity of $O(LNT)$, where L is the fixed value of 2000, our time complexity is lower since M is usually 50 steps and N is on average 10.

We also want to mention that the proposed approach is inherently a parallelizable optimization method (i.e. values of the mask, the element-wise multiplication, and the calculation of the gradient for each time step can be calculated in parallel) and is independent of any other time step values. This is in contrast to the dynamic programming approaches where the intermediate optimization values for each time step depend on the value of the previous time steps.

5. Experiments

5.1. Evaluation Protocols and Datasets

We evaluate FIFA on 3 different tasks: weakly supervised action segmentation, weakly supervised action alignment, and fully supervised action segmentation. We obtain the source code for the state-of-the-art approaches on each of these tasks and train a model using the standard training configuration of each model. Then we apply FIFA as a replacement for an existing inference stage or as an additional inference stage.

We evaluate our model using the **Breakfast** dataset [17] on the 3 different tasks. Breakfast is the most popular dataset currently used for action segmentation. It contains more than 1.7k videos of different cooking activities. The dataset consists of 48 different fine-grained actions. In our experiments, we follow the 4 train/test splits provided with the dataset and report the average. The main performance metrics used for weakly supervised action segmentation and alignment are the same as the previous approaches. The input features are also kept the same depending on the approach we use FIFA with.

5.2. Results and Discussions

Speed vs. Accuracy Trade-off One of the major benefits of FIFA is the flexibility of choosing the number of optimization steps. The number of steps of the optimization can be a tool to trade-off speed vs. accuracy. In exact inference, we can use frame-sampling i.e. lowering the resolution of the input features, or hypothesis pruning i.e. beam search for speed vs. accuracy trade-off.

Figure 2 plots the speed vs. accuracy trade-off of exact inference compared to FIFA. We observe that FIFA provides a much better speed-accuracy trade-off as compared to frame-sampling for exact inference. Table 1 analyzes the impact of the number of optimization steps on the accuracy and the associated computational overhead. As a result, the proposed approach achieves the best performance after 50 steps with 5.9% improvement on the MoF accuracy compared to not performing any inference. Moreover, it is more than 5 times faster than the exact inference.

Impact of the Length Energy For the length energy, we assume that the segment lengths follow a Laplace distribution. Figure 3 shows the impact of the length energy multiplier on the performance. While the best accuracy is achieved with a multiplier of 0.05, our approach is robust to the choice of these hyper-parameters. We further experimented with a Gaussian length energy. However, as shown in the figure, the performance is much worse compared to the Laplace energy. This is due to the quadratic penalty that dominates the total energy, which makes the optimization

Num. Steps	MoF	MoF-BG	IoU	IoD	Time (min)
No inference	45.4	44.7	37.3	51.2	1.0
2 steps	47.9	47.1	39.8	53.0	1.2
5 steps	49.1	48.3	40.0	52.8	1.5
10 steps	50.1	49.4	40.2	52.9	2.0
30 steps	51.2	50.6	41.0	53.2	4.2
50 steps	**51.3**	**50.7**	**41.1**	**53.3**	6.5
60 steps	**51.3**	**50.7**	**41.1**	**53.3**	7.7
Exact Inference	50.7	50.3	40.9	54.0	32.85

Table 1. Impact of the number of optimization steps for FIFA+MuCon for weakly supervised action segmentation on the Breakfast dataset.

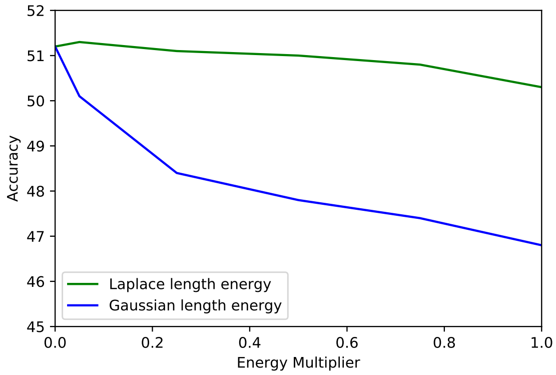

Figure 3. Effect of the length energy multiplier for Laplace and Gaussian length energy. Accuracy is calculated on the breakfast dataset using FIFA applied to the MuCon approach trained in the weakly supervised action segmentation setting.

Inference Method	initialization	MoF
Exact	MuCon [33]	**50.7**
	Equal	48.8 (-1.9)
FIFA	MuCon [33]	**51.3**
	Equal	50.2 (-1.1)

Table 2. Impact of the Length Model initialization for MuCon using exact inference and FIFA for weakly supervised action segmentation on the Breakfast dataset.

biased towards the initial estimate and ignores the observation energy.

Impact of Length Model Initialization Since FIFA starts with an initial estimate for the lengths, the choice of initialization might have an impact on the performance. Table 2 shows the effect of initializing the lengths with equal values compared to using the length model of MuCon [33] for the weakly supervised action segmentation on the Breakfast dataset. As shown in the table, FIFA is more robust to initialization compared to the exact inference as the drop in performance is half of the exact inference.

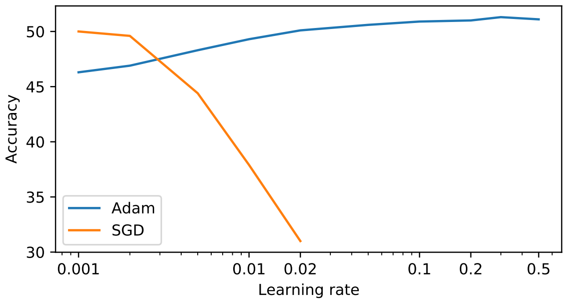

Figure 4. Effect of the learning rate on the performance of weakly supervised action segmentation using FIFA applied on the MuCon approach. Accuracy is calculated on the Breakfast dataset.

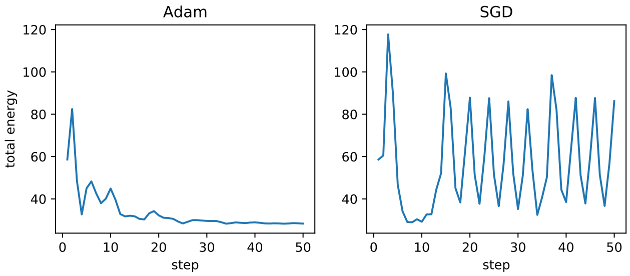

Figure 5. The value of the approximate energy during FIFA optimization for SGD and Adam optimizer for the same inference.

Optimizer and Its Learning Rate The choice of the optimizer used to update the length estimates using the calculated gradients is one of the hyper-parameters of our approach. We have experimented with two optimizers SGD and Adam. As shown in Figure 4, the best performing value for the learning rate hyper-parameter depends on the optimizer used. For SGD a low value of 0.001 achieves the best performance with higher values causing major drops in performance. On the other hand, Adam optimizer works well with a range of learning rate values as it has an internal mechanism to adjust the learning rate. The best performance for Adam is observed at 0.3.

We further investigate and notice that the reason SGD performs so poorly for large values of the learning rate is that it fluctuates and is not able to optimize the energy effectively. Figure 5 shows the value of the approximate energy during the optimization for Adam and SGD for the same inference. We observe that a large learning rate causes SGD to fluctuate while Adam is stable and achieves a lower energy value at the end of the optimization.

5.3. Comparison to State of the Art

In this section, we compare FIFA to other state-of-the-art approaches.

Weakly Supervised Action Segmentation We apply FIFA on top of two state-of-the-art approaches for weakly

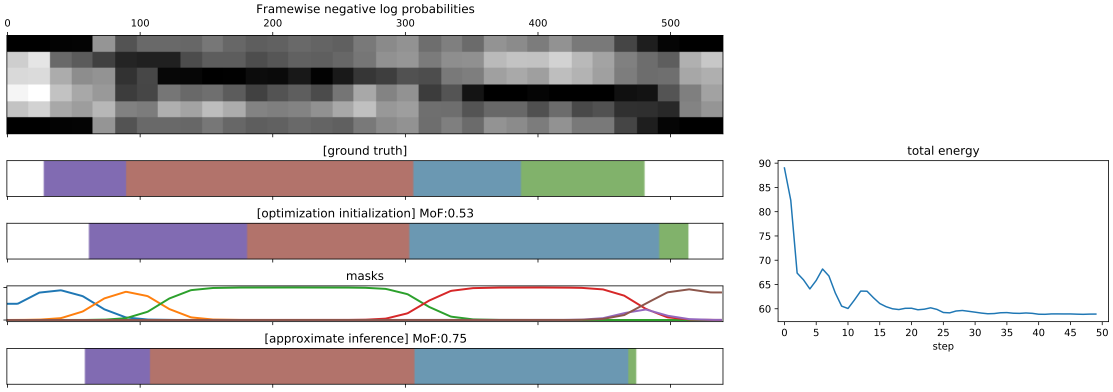

Figure 6. Visualization of the FIFA optimization process. On the right the values of the total approximate energy is plotted. On the left, negative log probability values, ground truth segmentation, optimization initialization, the masks and the segmentation after inference is ploted.

Method	MoF	MoF-BG	IoU	IoD	Time (min)
ISBA [6]	38.4	38.4	24.2	40.6	-
NNV [31]	43.0	-	-	-	-
D3TW [5]	45.7	-	-	-	-
CDFL [26]	50.2	48.0	33.7	45.4	-
CDFL*	49.4	47.5	35.2	46.4	260
FIFA + CDFL*	47.9	46.3	34.7	48.0	20.4 (×12.8)
MuCon [33]	47.1	-	-	-	-
MuCon*	50.7	50.3	40.9	**54.0**	4.1
FIFA + MuCon*	**51.3**	**50.7**	**41.1**	53.3	0.8 (×5.1)

Table 3. Results for weakly supervised action segmentation on the Breakfast dataset. * indicates results obtained by running the code on our machine.

supervised action segmentation namely MuCon [33] and CDFL [26] and report the results in Table3.

FIFA applied on CDFL achieves a 12 times faster inference speed while obtaining results comparable to exact inference. FIFA applied to MuCon achieves a 5 times faster inference speed and obtains a new state-of-the-art performance on the Breakfast dataset on most of the metrics.

Weakly Supervised Action Alignment Similar to weakly supervised action segmentation, we apply FIFA on top of CDFL and MuCon for weakly supervised action alignment task and report the results in Table 4. Our experiments show that FIFA applied on top of CDFL achieves state-of-the-art or better than state-of-the-art results on MoF and Mof-BG metrics, whereas FIFA applied on top of MuCon achieves state-of-the-art results for IoD and IoU metrics.

Fully Supervised Action Segmentation In the fully supervised action segmentation we apply FIFA on top of MS-TCN [1] and its variant MS-TCN++ [27] and report the results in Table 5. MS-TCN and MS-TCN++ are approaches that do not perform any inference at test time. This usually results in over-segmentation and low F1 and Edit scores. Applying FIFA on top of these approaches improves the F1 and Edit scores significantly. FIFA applied on top of MS-TCN achieves state-of-the-art performance and sets new state-of-the-art performance on most metrics.

Method	MoF	MoF-BG	IoU	IoD
ISBA [6]	53.5	51.7	35.3	52.3
D^{3}TW [5]	57.0	-	-	56.3
CDFL [26]	63.0	61.4	45.8	63.9
ADP [9]	64.1	**65.5**	43.0	-
FIFA + CDFL*	**65.3**	64.3	46.3	61.3
FIFA + MuCon*	61.4	61.2	**48.4**	**64.1**

Table 4. Results for weakly supervised action alignment on the Breakfast dataset.

Method	F1@{10, 25, 50}			Edit	MoF
BCN [36]	68.7	65.5	55.0	66.2	**70.4**
ASRF [14]	74.3	68.9	**56.1**	72.4	67.6
MS-TCN++ [27]	64.1	58.6	45.9	65.6	67.6
FIFA + MS-TCN++	74.3	69.0	54.3	77.3	67.9
MS-TCN [1]	52.6	48.1	37.9	61.7	66.3
FIFA + MS-TCN	**75.5**	**70.2**	54.8	**78.5**	68.6

Table 5. Results for fully supervised action segmentation setup on the Breakfast dataset.

5.4. Qualitative Example

A qualitative example of the FIFA optimization process is depicted in Figure 6. For further qualitative examples, failure cases, and details please refer to the supplementary material.

6. Conclusion

In this paper, we proposed FIFA a fast approximate inference procedure for action segmentation and alignment. Unlike previous methods, the proposed method does not rely on any expensive Viterbi decoding for inference. Instead, FIFA optimizes a differentiable energy function that can be minimized using gradient-descent which allow for a fast but also accurate inference during testing. We evaluated FIFA on top of fully and weakly supervised methods trained on the Breakfast dataset. The results show that FIFA is able to achieve comparable or better performance, while being at least 5 time faster than exact inference.

References

[1] Yazan Abu Farha and Juergen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. *CVPR*, 2019. 1, 2, 8

[2] David Belanger and Andrew McCallum. Structured prediction energy networks. In *ICML*, 2016. 2

[3] David Belanger, Bishan Yang, and Andrew McCallum. End-to-end learning for structured prediction energy networks. In *ICML*, 2017. 2

[4] Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Weakly supervised action labeling in videos under ordering constraints. In *ECCV*, 2014. 2

[5] Chien-Yi Chang, De-An Huang, Yanan Sui, Li Fei-Fei, and Juan Carlos Niebles. D^{3}TW: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. *CVPR*, 2019. 2, 8

[6] Li Ding and Chenliang Xu. Weakly-supervised action segmentation with iterative soft boundary assignment. In *CVPR*, 2018. 2, 8

[7] Olivier Duchenne, Ivan Laptev, Josef Sivic, Francis Bach, and Jean Ponce. Automatic annotation of human actions in video. In *ICCV*, 2009. 2

[8] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015. 2

[9] Reza Ghoddoosian, Saif Sayed, and Vassilis Athitsos. Action duration prediction for segment-level alignment of weakly-labeled videos. In *WACV*, 2021. 8

[10] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 2

[11] Cong Duy Vu Hoang, Gholamreza Haffari, and Trevor Cohn. Towards decoding as continuous optimization in neural machine translation. In *EMNLP*, 2017. 2

[12] De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *ECCV*, 2016. 2

[13] Yifei Huang, Yusuke Sugano, and Yoichi Sato. Improving action segmentation via graph-based temporal reasoning. In *CVPR*, 2020. 2

[14] Yuchi Ishikawa, Seito Kasai, Yoshimitsu Aoki, and Hirokatsu Kataoka. Alleviating over-segmentation errors by detecting action boundaries. In *WACV*, 2021. 2, 8

[15] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 2

[16] Svebor Karaman, Lorenzo Seidenari, and Alberto Del Bimbo. Fast saliency based pooling of fisher encoded dense trajectories. In *ECCV THUMOS Workshop*, 2014. 2

[17] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *CVPR*, 2014. 1, 6

[18] Hilde Kuehne, Juergen Gall, and Thomas Serre. An end-to-end generative framework for video segmentation and recognition. In *WACV*, 2016. 2

[19] Hilde Kuehne, Alexander Richard, and Juergen Gall. Weakly supervised learning of actions from transcripts. *Computer Vision and Image Understanding*, 163:78–89, 2017. 1, 2

[20] Hilde Kuehne, Alexander Richard, and Juergen Gall. A Hybrid RNN-HMM approach for weakly supervised temporal action segmentation. *PAMI*, 42(04):765–779, 2020. 2

[21] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *ICML*, 2014. 2

[22] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *CVPR*, 2017. 1, 2

[23] Colin Lea, Austin Reiter, René Vidal, and Gregory D Hager. Segmental spatiotemporal cnns for fine-grained action segmentation. In *ECCV*, 2016. 2

[24] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1, 2006. 2

[25] Peng Lei and Sinisa Todorovic. Temporal deformable residual networks for action segmentation in videos. In *CVPR*, 2018. 2

[26] Jun Li, Peng Lei, and Sinisa Todorovic. Weakly supervised energy-based learning for action segmentation. In *ICCV*, 2019. 1, 2, 3, 5, 8

[27] Shijie Li, Yazan Abu Farha, Yun Liu, Ming-Ming Cheng, and Juergen Gall. MS-TCN++: Multi-stage temporal convolutional network for action segmentation. *PAMI*, 2020. 1, 2, 8

[28] Davide Moltisanti, Sanja Fidler, and Dima Damen. Action recognition from single timestamp supervision in untrimmed videos. In *CVPR*, 2019. 5

[29] Hamed Pirsiavash and Deva Ramanan. Parsing videos of actions with segmental grammars. In *CVPR*, 2014. 2

[30] Alexander Richard, Hilde Kuehne, and Juergen Gall. Weakly supervised action learning with rnn based fine-to-coarse modeling. In *CVPR*, 2017. 1, 2

[31] Alexander Richard, Hilde Kuehne, Ahsan Iqbal, and Juergen Gall. Neuralnetwork-viterbi: A framework for weakly supervised video learning. In *CVPR*, 2018. 1, 2, 3, 8

[32] Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele. A database for fine grained activity detection of cooking activities. In *CVPR*, 2012. 2

[33] Yaser Souri, Mohsen Fayyaz, Luca Minciullo, Gianpiero Francesca, and Juergen Gall. Fast weakly supervised action segmentation using mutual consistency. In *arXiv*, 2019. 1, 2, 3, 5, 7, 8

[34] Yaser Souri, Alexander Richard, Luca Minciullo, and Juergen Gall. On evaluating weakly supervised action segmentation methods. In *arXiv*, 2020. 3

[35] Nam N Vo and Aaron F Bobick. From stochastic grammar to bayes network: Probabilistic parsing of complex activity. In *CVPR*, 2014. 2

[36] Zhenzhi Wang, Ziteng Gao, Limin Wang, Zhifeng Li, and Gangshan Wu. Boundary-aware cascade networks for temporal action segmentation. In *ECCV*, 2020. 1, 2, 8