# Uncertainty-Aware Representation Learning for Action Segmentation

**Lei Chen**[1] , **Muheng Li**[1] , **Yueqi Duan**[2] , **Jie Zhou**[1,*] , **Jiwen Lu**[1]
[1]Beijing National Research Center for Information Science and Technology (BNRist),
and the Department of Automation, Tsinghua University
[2]Department of Electronic Engineering, Tsinghua University
chenlei2020@mail.tsinghua.edu.cn, li-mh20@mails.tsinghua.edu.cn, duanyueqi@tsinghua.edu.cn,
jzhou@mail.tsinghua.edu.cn, lujiwen@tsinghua.edu.cn

## Abstract

In this paper, we propose an uncertainty-aware representation Learning (UARL) method for action segmentation. Most existing action segmentation methods exploit continuity information of the action period to predict frame-level labels, which ignores the temporal ambiguity of the transition region between two actions. Moreover, similar periods of different actions, e.g., the beginning of some actions, will confuse the network if they are annotated with different labels, which causes spatial ambiguity. To address this, we design the UARL to exploit the transitional expression between two action periods by uncertainty learning. Specially, we model every frame of actions with an active distribution that represents the probabilities of different actions, which captures the uncertainty of the action and exploits the tendency during the action. We evaluate our method on three popular action prediction datasets: Breakfast, Georgia Tech Egocentric Activities (GTEA), and 50Salads. The experimental results demonstrate that our method achieves the performance with state-of-the-art.

## 1 Introduction

Human action analysis has become a significant area in computer vision since the study of human movement is indispensable in practical application [Li *et al.*, 2021]. With the development of action analysis [Menapace *et al.*, 2021], the input data changes from videos posing in the conditional environment to the untrimmed videos in the wild [Sultani *et al.*, 2018]. The complexity of actions changes from a single step to a series of activities, such as the instructional video. Despite the enormous amount of works conducted in this area [Ahn and Lee, 2021; Wang *et al.*, 2020; Li *et al.*, 2020; Ishikawa *et al.*, 2021], the tasks in the action analysis are still challenging, such as action segmentation. Because the pattern diversity of activities is very rich and the lengths variances of actions are considerable. Detecting the actions in a long video is to localize the start time and the terminal time in the whole video for every existing action [Dai *et al.*, 2021]. Unlike the action detection task, action segmentation aims to predict the correct category of every frame in the video, which obtains the start time and the terminal time of actions and segments the actions in the frame level [Sridhar *et al.*, 2021]. The action segmentation is defined as predicting the action category of every frame and evaluated by the prediction accuracy of all frames [Lea *et al.*, 2017a].

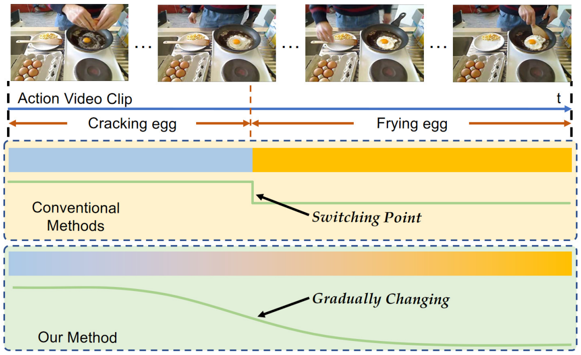


Figure 1: Comparisons of the proposed UARL with conventional action segmentation methods. The top row shows an example of an action video clip that contains two actions. The provided annotations are 'cracking egg' and 'frying egg' in the temporal domain. The conventional action segmentation methods treat the annotations into one-hot vectors to supervise the training process of the segmentation networks. However, this manner creates a switching point between two actions. The probability of action 'cracking egg' changes smoothly in the temporal domain. We propose the uncertainty-aware representation learning method to encode the gradually changing tendency. (Best viewed in color)

It is challenging to accurately segment the actions at the frame level, especially for the transition region between two actions. The majority of existing methods for action segmentation can be mainly classified into three categories: methods of the sequential model [Farha and Gall, 2019], architectures of multi-stream [Gammulle *et al.*, 2019], and temporal convolutional networks [Lei and Todorovic, 2018]. The methods of the first category aim to capture the long-short term relation with the iterative framework. Still, they are neither flexible for the input data nor remembering the information of all time. The methods for the second category extract complementary information from different streams of input data, such as RGB frames and depth images. The compu-

[*] Corresponding author.

tational cost of these methods increases with the number of different streams. For the third category, the methods design framework with Temporal Convolutional Networks (TCNs). These methods adjust handcrafted receptive fields of the network, which cannot satisfy the requirement of generalization. All these methods ignore a significant point: the input data's ambiguity. For example, in Figure 1, ambiguity exists in the transition region between two action periods. The annotated action categories switch suddenly at the transformation frame. However, the changing process of actions lasts several frames. This kind of annotation creates ambiguity and makes it challenging for action segmentation.

To address the above limitations of previous works for action segmentation, we present an uncertainty-aware representation learning (UARL) method. Figure 1 shows the motivation of our approach. Our UARL aims to model the ambiguity of frames by uncertainty learning. We consider that the probabilities of frames between two actions should change smoothly from one activity to another instead of either abrupt change. The smooth changing of probability can reduce the interference from the annotated frames. To describe and eliminate the annotation ambiguity of action frames, we model every frame of the input video sequence into a distribution represented by the parameterized Gaussian distribution. Through this soft supervision, our model can learn a mapping from the input RGB data into a probability space. Then we utilize the sampling operation to reflect the probability of action and predict the action category for every frame. Experimental results on three benchmarks demonstrate the effectiveness of our proposed approach.

## 2 Related Work

**Action Segmentation:** Many works have been proposed for action segmentation. These previous methods can be mainly classified into three categories: sequential model methods, multi-stream architectures, and temporal convolutional networks. For the first category, the proposed sequential models exploit the long-short term dependencies, which apply in the iterative architecture. [Farha and Gall, 2019] proposed a multi-stage architecture to directly predict the label of every frame by utilizing the temporal convolutions. Each stage generates a candidate prediction for the next refine step, trained with the classification loss and smooth loss. But these models are limited by the inflexible modules and suffer from information forgetting. For the second category, multi-stream approaches utilize complementary information among different streams. [Gammulle *et al.*, 2019] presented a CGAN model to continuously fine-grain human action segmentation, using the RGB frames and the complementary information, such as optical flow and depth data. But these methods increase the computational redundancy in capturing long-short term information of multiple streams. For the third category, these methods are based on the temporal convolutional network (TCN), a unified structure by adjusting receptive fields and processing long videos in parallel. [Lei and Todorovic, 2018] proposed an approach for computing two parallel temporal streams: residual stream for full temporal resolution and pooling/un-pooling steam for capturing long-range video information. But these adjustments of receptive fields still rely on human design, which is not appropriate.

**Uncertainty Learning:** In recent years, uncertainty learning has made obvious progress in computer vision and attracted more attention in modeling the ambiguity tasks by improving discriminant Deep Neural Networks (DNNs). There are two main categories of uncertainty learning methods: model uncertainty and data uncertainty. For the first category, model uncertainty represents the uncertainty of model parameters caused by the training data [MacKay, 1992; Kendall *et al.*, 2015; Gal and Ghahramani, 2016]. This kind of uncertainty can be reduced by enlarging the number of data. [Kendall *et al.*, 2015] proposed a deep-learned method of probabilistic pixel-wise segmentation that can predict pixel-wise category with a measure of model uncertainty. They conducted Monte-Carlo sampling with a dropout at the testing process to generate a posterior distribution of pixel-level categories. [Gal and Ghahramani, 2016] proposed a framework casting dropout training in Deep Neural Networks (DNNs) by approximating Bayesian inference in deep Gaussian processes, which can model uncertainty with drop neural networks. The proposed method represents the uncertainty without sacrificing either computational complexity or accuracy. For the second category, data uncertainty considers the uncertainty in the input data caused by the process of capturing data [Kendall and Gal, 2017; Kingma and Welling, 2014]. This kind of uncertainty comes from the noise in the original data and cannot be eliminated by simply enlarging the amount of data. [Kingma and Welling, 2014] showed that the reparameterization of the variational lower bound yields a lower bound estimator for straightforwardly optimizing standard stochastic gradient methods. They proposed a stochastic variational inference and learning architecture, which scales to large datasets under mild differentiable conditions. [Kendall and Gal, 2017] studied models under the framework for two tasks: per-pixel semantic segmentation and depth estimation by presenting an explicit uncertainty formulation. The proposed formulation can be interpreted as learned attenuation. In this work, we propose an uncertainty-aware representation learning (UARL) method for action segmentation by representing the uncertainty of every frame.

## 3 Proposed Approach

In this section, we first introduce the pipeline of uncertainty-aware representation learning (UARL). Then we present the details of our method.

Figure 2 shows the pipeline of our proposed UARL. The input of our method is the untrimmed video, and we handle every frame recurrently, which aims to encode the temporal relation between adjacent frames. Firstly, we extract the frame-level feature and utilize the extracted feature as the input of the uncertainty prediction module. Then we predict several results of action categories with uncertainty prediction shown at the bottom of Figure 2. Lastly, we fuse the predicted action classes to refine the final prediction of the input frame. The bottom part of Figure 2 illustrates the details of the process of uncertainty prediction. We utilize the

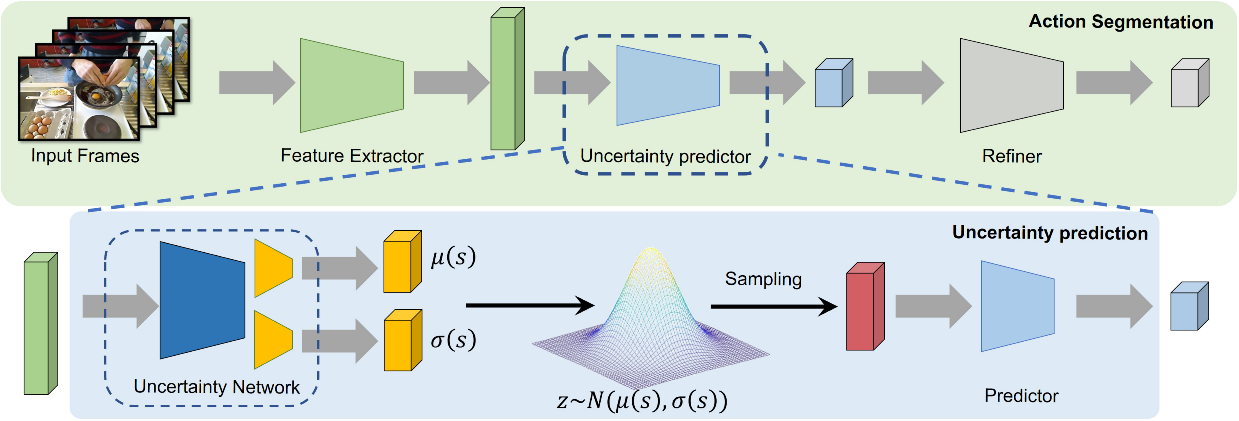


Figure 2: The pipeline of our method and details of uncertainty learning. We utilize the feature extractor to capture the original features of input frames. Then we predict the uncertainty of the frames by mapping the original features into the corresponding distributions. We employ the Monte-Carlo sampling operation to obtain the uncertainty feature and predict the action category with the predictor. Lastly, we refine the category of the input image with the predicted action category that has encoded the uncertainty of the input action.

extracted feature to establish the distribution of current time and consider the consistency in the temporal domain [Guan *et al.*, 2021]. Our uncertainty network establishes a mapping from the feature to a distribution. Then we sample from the learned distribution to generate the sampling feature for predicting one probable action for the action segmentation.

**Feature Extractor:** We define the input sequence of the video as $\mathbf{S} = \{\mathbf{s}_t : 1, ..., T\} \in R^{n_w \times n_h}$, where $\mathbf{s}_t$ refers to the $t$th frame. $T$ is the total length of the video. $n_w$ and $n_h$ are the width and height of the input frames. We utilize the feature extractor, $\Omega(\cdot)$, to capture the original features from the video that is defined as $\mathbf{F} = \{\mathbf{f}_t : 1, ..., T\}$, which is formulated as $\mathbf{F} = \Omega(\mathbf{S})$. The extracted feature $\mathbf{F}$ contains the spatio-temporal information since we use the spatial-temporal convolutional network as the backbone of the feature extractor.

**Uncertainty Predictor:** The annotation of every frame is a one-hot vector which means that every frame only represents one specific action, obviously imprecise for the classifier. For example, at the beginning of an action, the observed frames may be several possible actions, which are reflected by the uncertainty, in other words. We design the uncertainty predictor to reduce the interference of the rigid annotation for the network. We represent the uncertainty of input videos as probabilistic distributions formulated as $z \sim p(z|\mathbf{f}_t)$. Since we predict the uncertainty for every frame, we only mark the input and output variables with frame index $t$ at the subscript. For the input $\mathbf{f}_t$ and the corresponding label $y$, we marginalize over the embedding distribution to compute the corresponding probability as the prediction that is represented as follows:

$$p(y|\mathbf{f}_t) = \int p(y|z)p(z|\mathbf{f}_t)dz, \tag{1}$$

However, this representation is difficult for the designed network to utilize and train. We approximate the (1) with the Monte-Carlo sampling, which is represented as follows:

$$p(y|\mathbf{f}_t) \approx \frac{1}{N}\sum_{n=1}^{N} p(y|z^{(n)}), \tag{2}$$

From (2), the total number of sampling times equals $N$ for every feature $\mathbf{f}_t$. $n$ refers to the current sampling times.

We define the embedding $z$ as a multivariate Gaussian distribution and sample from the distribution $p(z|\mathbf{f}_t)$, which can be formulated as follows:

$$p(z|\mathbf{f}_t) = \mathbb{N}(z; \boldsymbol{\nu}(\mathbf{f}_t), \boldsymbol{\sigma}(\mathbf{f}_t)), \tag{3}$$

where $\boldsymbol{\nu}(\mathbf{s}$ is the mean vector and $\boldsymbol{\sigma}(\mathbf{s})$ represents the diagonal covariance. We denote the parameters of the shared layers in the uncertainty network as $\delta_s(\cdot)$ and denote corresponding parameters of the head branches network as $\delta_\nu(\cdot)$ and $\delta_\sigma(\cdot)$. The mean and diagonal covariance are represented as follows:

$$\boldsymbol{\nu}(\mathbf{f}_t) = \delta_\nu(\delta_s(\mathbf{f}_t)), \boldsymbol{\sigma}(\mathbf{f}_t) = \delta_\sigma(\delta_s(\mathbf{f}_t)), \tag{4}$$

where $\boldsymbol{\nu}(\mathbf{s})$ reflects the most likely probability that decides the action category of the current frame. $\boldsymbol{\sigma}(\mathbf{s})$ reflects the intensity of the embedding variance that represents the diversity of the action. The more the variance is, the more the uncertainty of the input sequence is, and the more diverse the current frame is. The sampling operation cannot be differentiable. Thus we utilize the reparameterization trick mentioned in [Kingma and Welling, 2014] to make the operation backpropagate as follows:

$$z^{(n)} = \boldsymbol{\nu}(\mathbf{f}_t) + \text{diag}(\sqrt{\boldsymbol{\sigma}(\mathbf{f}_t)}) \cdot \pi^{(n)}, \pi^{(n)} \sim \mathbb{N}(\mathbf{0}, \boldsymbol{I}), \tag{5}$$

Based on (5), we sample noise from the Gaussian distribution of $\mathbb{N}(\mathbf{0}, \boldsymbol{I})$. Then we can obtain the $z^{(n)}$ with (5) instead of sampling from the original distribution of $\mathbb{N}(\boldsymbol{\nu}(\mathbf{f}_t), \boldsymbol{\sigma}(\mathbf{f}_t))$. With this format, the network can easily back-propagate in the training process. Our proposed uncertainty module aims to decrease the disturbance of the hard annotation, especially for the transition region, in the training process. Sampling operation provides us the embedded feature that is denoted as $u^{(n)}$ and is formulated as follows:

$$u^{(n)} = \boldsymbol{\nu}(\mathbf{f}_t) + \text{diag}(\sqrt{\boldsymbol{\sigma}(\mathbf{f}_t)}) \cdot \phi(\pi^{(n)}), \pi^{(n)} \sim \mathbb{N}(\mathbf{0}, \boldsymbol{I}), \tag{6}$$

where $\phi(\cdot)$ is the sampling operation that refers to the sampling function in (2). We design another predictor network to generate the uncertainty prediction for the corresponding samplings. (6) is the format of the sampling process, which is derivable for the network training. For the $n$th sampling of the input frame $\mathbf{f}_t$, we denote the prediction result as $\mathcal{A}_t^{(n)}$. Based on the previous formulation (4) and (6), the uncertainty prediction $\mathcal{A}_t^{(n)}$ can be represented as follows:

$$\begin{aligned}\mathcal{A}_t^{(n)} &= \omega\Phi(u^{(n)}) = \omega\Phi[\boldsymbol{\nu}(\mathbf{f}_t) + \mathrm{diag}(\sqrt{\boldsymbol{\sigma}(\mathbf{f}_t)}) \cdot \phi(\pi^{(n)})] \\ &= \omega\Phi[\delta_\nu(\delta_s(\mathbf{f}_t)) + \mathrm{diag}(\sqrt{\delta_\sigma(\delta_s(\mathbf{f}_t))}) \cdot \phi(\pi^{(n)})], \end{aligned} \tag{7}$$

where $\Phi$ refers to the parameters of the embedding layers in the predictor network and $\omega = [\omega_1, \omega_2, ..., \omega_C]^\top$ is the parameters of the classification layer. $C$ is the total category of actions in videos. Thus we integrate the existing classification methods with our proposed approach, and the loss function is formulated as follows:

$$\begin{aligned}\mathcal{L}_{pre}(\mathbf{f}_t) &= -\frac{1}{NT}\sum_{t=1}^{T}\sum_{n=1}^{N}\log p(\mathcal{A}_t^{(n)} = c|z^{(n)}; \delta_{s,\nu,\sigma}, \Phi, \omega, \phi) \\ &= -\frac{1}{NT}\sum_{t=1}^{T}\sum_{n=1}^{N}\log\frac{\exp(\omega_c^\top\Phi(u^{(n)}))}{\sum_{r=1}^{C}\exp(\omega_r^\top\Phi(u^{(n)}))}. \end{aligned} \tag{8}$$

Moreover, we tend to drop the discriminative information as little as possible. Thus we design a reconstruction loss $\mathcal{L}_{rec}$, which computes the difference between the sampling and original features and can be represented as $\mathcal{L}_{rec} = \|u^{(n)} - \mathbf{f}_t\|_1$. Moreover, since the consecutive frames in videos have temporal continuity, the uncertainty of consecutive frames should also be close. We propose an extra uncertainty smoothing loss $\mathcal{L}_{s-unc}$ together with the smoothing loss of predicted labels $\mathcal{L}_{s-pre}$ to guarantee the frame-wise prediction continuity.

$$\mathcal{L}_{s-pre}(\mathbf{f}_t) = \frac{1}{NT}\sum_{t=1}^{T}\sum_{n=1}^{N}(\mathcal{A}_{t-1}^{(n)} - \mathcal{A}_t^{(n)})^2 \tag{9}$$

$$\mathcal{L}_{s-unc}(\mathbf{f}_t) = \frac{1}{T}\sum_{t=1}^{T}\|\boldsymbol{\sigma}(\mathbf{f}_{t-1}) - \boldsymbol{\sigma}(\mathbf{f}_t)\|_2 \tag{10}$$

To avoid the predictions from degenerating into deterministic embeddings by outputting negligible uncertainties, we introduced an extra regularization loss term $\mathcal{L}_{reg}$ to bound the learned distribution from standard normal distributions, which is a KL divergence term as follows:

$$\mathcal{L}_{reg}(\mathbf{f}_t) = KL(\mathbb{N}(z; \boldsymbol{\nu}(\mathbf{f}_t), \boldsymbol{\sigma}(\mathbf{f}_t))\|\mathbb{N}(\mathbf{0}, \mathbf{I})). \tag{11}$$

The final loss function for training the module of uncertainty prediction is denoted as $\mathcal{L}_{un}$ that is formulated as follows:

$$\mathcal{L}_{un} = \mathcal{L}_{pre} + \lambda_1\mathcal{L}_{rec} + \lambda_2\mathcal{L}_{s-pre} + \lambda_3\mathcal{L}_{s-unc} + \lambda_4\mathcal{L}_{reg}. \tag{12}$$

**Refiner:** We design a refiner to make the prediction result for frames more smooth. Our proposed uncertainty prediction architecture generates the potential action category frame by frame. The predicted results may have the situation that the current prediction is a trip point. This point is unreasonable for a sequence prediction since temporal consistency is a common constraint for sequential tasks. Thus we design a refiner module to combine the consistency of actions and the result of uncertainty prediction. The output of the uncertainty prediction is based on the sampling times $n$, which should be fused to predict the one most probable action category. We utilize the vote operation and formulate the prediction as $\mathcal{C}_t = \Lambda(\mathcal{A}_t^{(n)})$. We denote the parameters of the refiner network as $\theta(\cdot)$, and the output can be represented as follows:

$$\mathbf{M}_t = \theta(\mathcal{C}_t) = \theta(\Lambda(\mathcal{A}_t^{(n)})), \tag{13}$$

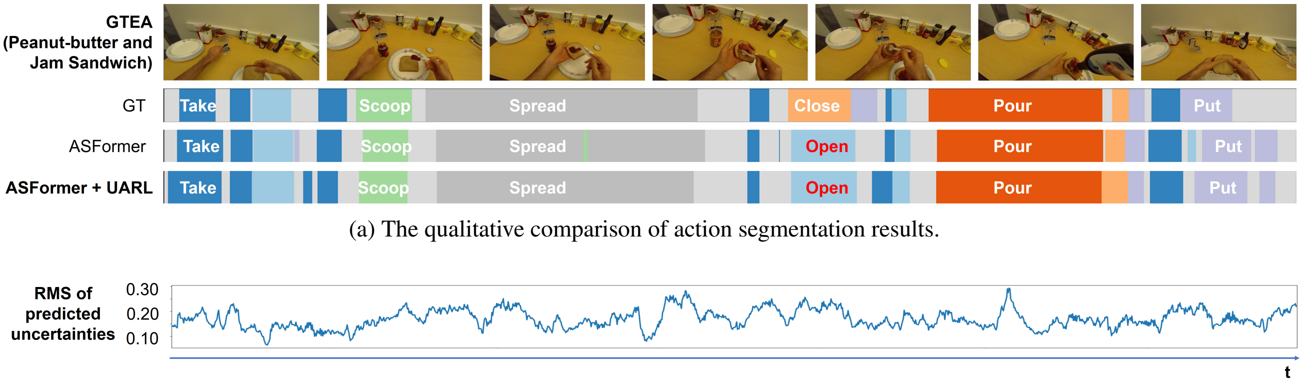
## 4 Experiments

In this section, we first introduce three datasets that we conducted the experiments on. Then we describe the implementation details of our model. Lastly, we show our experiments and comparisons with other methods on three datasets.

**Datasets:** We evaluate our proposed method on three challenging action segmentation datasets: 50Salads [Stein and McKenna, 2013], Georgia Tech Egocentric Activities (GTEA) [Fathi *et al.*, 2011], and the Breakfast dataset [Hilde *et al.*, 2014]. The 50Salads dataset contains 50 top-view videos of salad preparation with 17 different action classes. The 5-fold cross-validation is performed for evaluation. The GTEA dataset contains 28 egocentric videos of daily kitchen activities with 11 classes of action. We use the 4-fold cross-validation to evaluate the performances. The Breakfast dataset contains 1,712 third-person videos of breakfast preparation activities, including 48 types of different actions. For evaluation, the standard 4-fold cross-validation is used to evaluate the performances. For consistency, all videos from the above datasets are set to 15 fps.

**Implementation Details:** The UARL approach is implemented as an implanted module plugged into the existing action segmentation backbone models. For the inputs, we used the 2048-d I3D [Carreira and Zisserman, 2017] frame-wise features extracted by [Farha and Gall, 2019]. We chose 3 different state-of-the-art backbone models to validate the effectiveness of our approach: MS-TCN++ [Li *et al.*, 2020], ASRF [Ishikawa *et al.*, 2021], and ASFormer [Yi *et al.*, 2021]. The backbone model plays the roles of the feature extractor and the refiner in our proposed pipeline. The extracted feature size is set to 64 for each video frame. The head branches network $\delta_\nu$ and $\delta_\sigma$ are implemented by 1D convolutions. The total sample time $N$ is dynamically adjusted according to model scales and video lengths. For the loss function, we set the weight as $0.01$ for $\mathcal{L}_{rec}$ ($\lambda_1$), the weight of the regularization term is set to $1e^{-4}$ ($\lambda_4$), and the weights of all smoothing terms are set to $0.15$ ($\lambda_2$ and $\lambda_3$). We trained the models for 200 epochs. For optimization, we used Adam optimizer with the learning rate of $0.0005$ and the batch size of 1 for all experiments in this paper.

**Evaluation Metrics:** For evaluation, we use three metrics: the frame-wise accuracy (Acc), segmental edit distance, and the segmental F1 score at overlapping thresholds 10%, 25%, and 50%, denoted by F1@10,25,50. While frame-wise accuracy is the most common metric, it is insensitive to over-segmentation errors. To better penalize for over-segmentation errors, we also choose the segmental edit distance [Lea *et al.*, 2016] and segmental F1 score [Lea *et al.*, 2017b].



(a) The qualitative comparison of action segmentation results.

(b) The root mean square result of the predicted uncertainties for the sample in (a).

Figure 3: The illustration of the action segmentation results.

| GTEA | F1 Score | | | Edit | Acc |
|---|---|---|---|---|---|
| | @10 | @25 | @50 | | |
| MSTCN++ | 88.8 | 85.7 | 76.0 | 83.5 | 80.1 |
| MSTCN++ + UARL | 90.1 | 87.8 | 76.5 | 84.9 | 78.8 |
| ASRF | 89.4 | 87.8 | 79.8 | 83.7 | 77.3 |
| ASRF + UARL | 89.9 | 88.0 | 75.5 | 86.2 | 77.3 |

| 50 Salads | F1 Score | | | Edit | Acc |
|---|---|---|---|---|---|
| | @10 | @25 | @50 | | |
| MSTCN++ | 80.7 | 78.5 | 70.1 | 74.3 | 83.7 |
| MSTCN++ + UARL | 80.8 | 78.7 | 69.5 | 74.6 | 82.7 |
| ASRF | 84.9 | 83.5 | 77.3 | 79.3 | 84.5 |
| ASRF + UARL | 85.1 | 83.6 | 77.4 | 78.7 | 84.0 |

| Breakfast | F1 Score | | | Edit | Acc |
|---|---|---|---|---|---|
| | @10 | @25 | @50 | | |
| MSTCN++ | 64.1 | 58.6 | 45.9 | 65.6 | 67.6 |
| MSTCN++ + UARL | 65.2 | 59.4 | 47.4 | 66.2 | 67.8 |

Table 1: Improvement results on action segmentation dataset.

| Method | F1 Score | | | Edit | Acc |
|---|---|---|---|---|---|
| | @10 | @25 | @50 | | |
| BCN | 88.5 | 87.1 | 77.3 | 84.4 | 79.8 |
| MS-TCN++ | 88.8 | 85.7 | 76.0 | 83.5 | **80.1** |
| ASRF | 89.4 | 87.8 | 79.8 | 83.7 | 77.3 |
| G2L | 89.9 | 87.3 | 75.8 | 84.6 | 78.5 |
| SSTDA | 90.0 | 89.1 | 78.0 | 86.2 | 79.8 |
| SSTDA+HASR | 90.9 | 88.6 | 76.4 | 87.5 | 78.7 |
| ASFormer | 90.1 | 88.8 | 79.2 | 84.6 | 79.7 |
| ASRF + UARL | 89.9 | 88.0 | 75.5 | 86.2 | 77.3 |
| ASFormer + UARL | **92.7** | **91.5** | **82.8** | **88.1** | 79.6 |

Table 2: Comparing our approach with SOTA methods on GTEA.

| Method | F1 Score | | | Edit | Acc |
|---|---|---|---|---|---|
| | @10 | @25 | @50 | | |
| MS-TCN++ | 80.7 | 78.5 | 70.1 | 74.3 | 83.7 |
| BCN | 82.3 | 81.3 | 74.0 | 74.3 | 84.4 |
| SSTDA | 83.0 | 81.5 | 73.8 | 75.8 | 83.2 |
| ASRF | 84.9 | 83.5 | 77.3 | 79.3 | 84.5 |
| ASFormer | 85.1 | 83.4 | 76.0 | 79.6 | 85.6 |
| ASFormer + ASRF | 85.1 | 85.4 | **79.3** | **81.9** | **85.9** |
| SSTDA + HASR | **86.6** | **85.7** | 78.5 | 81.0 | 83.9 |
| MSTCN++ + UARL | 80.8 | 78.7 | 69.5 | 74.6 | 82.7 |
| ASRF + UARL | 85.3 | 83.5 | 77.8 | 78.2 | 84.1 |

Table 3: Comparing our approach with SOTA methods on 50 Salads.

**Improvements from State-of-the-Art Models:** To validate that our UARL can boost the performance of existing models for action segmentation, we plugged in our UARL to the backbone models mentioned above on the three action segmentation datasets. The results are shown in Table 1. Our approach generally improves the action segmentation performances from the backbone models. Specifically, UARL mainly improves performance by evaluating F1 score and segmental edit distance, which demonstrates that the uncertainty-aware module can improve the backbone models by reducing over-segmentation errors. Figure 3(a) displays the qualitative comparison of the action segmentation results. It can be seen that some of the transient segmentation errors have been corrected by our approach, which indicates that the uncertainty-aware method has a better ability to represent the indeterminate frames.

**Comparison with the State-of-the-Art:** By introducing the uncertainty-aware module into the action segmentation backbone models, our approach achieves competitive performances of the current state-of-the-art. We compare our method with various action segmentation approaches, including BCN [Wang *et al.*, 2020], MS-TCN++ [Li *et al.*, 2020], ASRF [Ishikawa *et al.*, 2021], G2L [Gao *et al.*, 2021], SSTDA [Chen *et al.*, 2020], HASR [Ahn and Lee, 2021], and ASFormer [Yi *et al.*, 2021]. Table 2 shows the comparison results on the GTEA dataset. Our approach (ASFormer + UARL) reaches the current state-of-the-art except for the Acc metric. Moreover, Table 3 shows the comparison results on the 50 Salads dataset. Our approach (ASRF + UARL) achieves competitive performances with current state-of-the-art methods. It should be noticed that ASRF is not the cur-

| Loss terms | | | | | F1 Score | | | Edit | Acc |
|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{L}_{pre}$ | $\mathcal{L}_{s-pre}$ | $\mathcal{L}_{reg}$ | $\mathcal{L}_{rec}$ | $\mathcal{L}_{s-unc}$ | @10 | @25 | @50 | | |
| √ | √ | × | × | × | 88.5 | 86.7 | 74.8 | 82.4 | 79.1 |
| √ | √ | √ | × | × | 88.9 | 87.3 | 73.7 | 83.6 | 80.0 |
| √ | √ | √ | √ | × | 89.1 | 87.2 | 74.3 | 84.1 | 77.5 |
| √ | √ | √ | √ | √ | 90.1 | 87.8 | 76.5 | 84.9 | 78.8 |

Table 4: Effect on different loss terms. The experiments are conducted on GTEA using MSTCN++ as the backbone model.

| GTEA | with $\mathcal{L}_{rec}$ | F1 Score | | | Edit | Acc |
|---|---|---|---|---|---|---|
| | | @10 | @25 | @50 | | |
| ASFormer | × | 91.8 | 90.8 | 81.6 | 87.2 | 81.3 |
| | √ | 92.7 | 91.5 | 82.8 | 88.1 | 79.6 |

| 50 Salads | with $\mathcal{L}_{rec}$ | F1 Score | | | Edit | Acc |
|---|---|---|---|---|---|---|
| | | @10 | @25 | @50 | | |
| MSTCN++ | × | 78.2 | 76.1 | 67.9 | 70.4 | 83.2 |
| | √ | 80.8 | 78.7 | 69.5 | 74.6 | 82.7 |

| Breakfast | with $\mathcal{L}_{rec}$ | F1 Score | | | Edit | Acc |
|---|---|---|---|---|---|---|
| | | @10 | @25 | @50 | | |
| MSTCN++ | × | 62.3 | 56.7 | 44.3 | 64.6 | 66.7 |
| | √ | 65.2 | 59.4 | 47.4 | 66.2 | 67.8 |

Table 5: Effect of the reconstruction loss term $\mathcal{L}_{rec}$.

rent state-of-the-art method. ASFormer is more competitive than ASRF, but it is not efficient since it uses the Transformer model [Vaswani *et al.*, 2017] as its backbone. The average length of the videos in 50Salads is longer than that in GTEA. Thus, the multiple-sampling strategy is not so applicable for the 50Salads dataset when using heavy backbone models. Even so, the UARL module still boosts the performance of previous state-of-the-art approaches.

**Effect of different loss terms:** Our UARL approach includes five loss terms in total. The $\mathcal{L}_{pre}$ and $\mathcal{L}_{s-pre}$ are modified from the original action segmentation loss terms [Farha and Gall, 2019] of the backbone models. Besides, we propose the other three-loss terms on the uncertainty-aware module. We perform several ablation studies to verify the effectiveness of the proposed losses. The results are displayed in Table 4. The regularization term $\mathcal{L}_{reg}$ improves the performances from original losses except for the F1@50 metric, which verifies the importance of preventing the model from outputting worthless uncertainties. The reconstruction loss $\mathcal{L}_{rec}$ is introduced to drop the discriminative information as much as possible, minimizing the L1 distance between the raw and embedded features. The introduction of $\mathcal{L}_{rec}$ helps boost the performances and speed up the convergence. The reconstruction loss makes the head branches networks $\delta_{\nu}$ and $\delta_{\sigma}$ as decomposition networks that split the frame-wise features into the mean and diagonal covariance of the multivariate Gaussian distribution. To further verify the effects of the reconstruction loss, we conduct additional ablation studies across different datasets and backbone models. The results are shown in Table 5. As can be seen, the reconstruction loss boosts the action segmentation performances in all the cases. We also test the effect on $\mathcal{L}_{s-unc}$. The smoothing loss in the original loss terms $\mathcal{L}_{s-pre}$ aims to constrain the final predicted labels to be as coherent as possible and control the over-segmentation problems. At the same time, $\mathcal{L}_{s-unc}$ bounds the temporal continuity of predicted uncertainties to represent the predicted distributions of adjacent frames better. The results prove that $\mathcal{L}_{s-unc}$ increases the final segmentation performance.

| Total sample time of inference | F1 Score | | | Edit | Acc |
|---|---|---|---|---|---|
| | @10 | @25 | @50 | | |
| $N = 10$ | 89.99 | 87.76 | 75.94 | 84.69 | 78.60 |
| $N = 20$ | 89.70 | 87.84 | 76.22 | 85.25 | 78.84 |
| $N = 30$ | 89.88 | 88.04 | 76.07 | 84.83 | 79.51 |
| $N = 40$ | 89.75 | 87.73 | 75.24 | 84.63 | 77.90 |
| $N = 50$ | 90.05 | 87.82 | 76.49 | 84.89 | 78.81 |

Table 6: Effect of the total sample time of inference stage. The experiments are conducted on GTEA using MSTCN++ as the backbone model.

**Effect of total sample time:** Our UARL approach adopts a dynamically adjusted sample time according to scales of backbone models and sizes of video samples. For example, regarding the heavy backbone models such as ASFormer [Yi *et al.*, 2021], we use fewer samples ($N < 20$). More samples can be collected ($N > 40$) for lighter backbone models such as MS-TCN++ [Li *et al.*, 2020]. Intuitively, a larger sample time contributes to better segmentation performances. We conduct ablation studies on the effect of sample time during the inference stage, and the results are displayed in Table 6. In general, the segmentation performance increases when more samples are being collected.

## 5 Conclusion

In this paper, we have proposed an uncertainty-aware representation learning (UARL) method for action segmentation. Our UARL aims to reduce the interference of the frame-level hard annotation in predicting the action of the transition region. We designed an uncertainty predictor to estimate the ambiguity of every frame. By uncertainty learning, we reflected the change in the probability of actions with the change on the corresponding distribution of actions to alleviate the influence of ambiguity in predicting actions in the temporal domain. We utilized the sampling operation to promise the module derivable and designed the reconstruction loss to preserve the temporal consistency of actions, which are two criteria of uncertainty prediction. Our proposed UARL method exploits the transitional expression between two action periods by modeling every frame of actions with a distribution. The experimental results on three datasets demonstrate the effectiveness of UARL in alleviating ambiguity and action segmentation.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant U1813218 and Grant 62125603, in part by a grant from the Beijing Academy of Artificial Intelligence (BAAI), and in part by the Postdoctoral Innovative Talent Support Program of China under Grant BX2021160.

## References

[Ahn and Lee, 2021] Hyemin Ahn and Dongheui Lee. Refining action segmentation with hierarchical video representations. In *ICCV*, pages 16302–16310, 2021.

[Carreira and Zisserman, 2017] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017.

[Chen *et al.*, 2020] Min-Hung Chen, Baopu Li, Yingze Bao, Ghassan AlRegib, and Zsolt Kira. Action segmentation with joint self-supervised temporal domain adaptation. In *CVPR*, pages 9454–9463, 2020.

[Dai *et al.*, 2021] Rui Dai, Srijan Das, and François Bremond. Learning an augmented rgb representation with cross-modal knowledge distillation for action detection. In *ICCV*, pages 13053–13064, 2021.

[Farha and Gall, 2019] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *CVPR*, June 2019.

[Fathi *et al.*, 2011] Alireza Fathi, Xiaofeng Ren, and James M Rehg. Learning to recognize objects in egocentric activities. In *CVPR*, pages 3281–3288, 2011.

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

[Gammulle *et al.*, 2019] Harshala Gammulle, Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. Coupled generative adversarial network for continuous fine-grained action segmentation. In *WACV*, pages 200–209, 2019.

[Gao *et al.*, 2021] Shang-Hua Gao, Qi Han, Zhong-Yu Li, Pai Peng, Liang Wang, and Ming-Ming Cheng. Global2local: Efficient structure search for video action segmentation. In *CVPR*, pages 16805–16814, 2021.

[Guan *et al.*, 2021] Dayan Guan, Jiaxing Huang, Aoran Xiao, and Shijian Lu. Domain adaptive video segmentation via temporal consistency regularization. In *ICCV*, pages 8053–8064, 2021.

[Hilde *et al.*, 2014] Kuehne Hilde, Bilgin Arslan Ali, and Serre Thomas. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *CVPR*, pages 780–787, 2014.

[Ishikawa *et al.*, 2021] Yuchi Ishikawa, Seito Kasai, Yoshimitsu Aoki, and Hirokatsu Kataoka. Alleviating over-segmentation errors by detecting action boundaries. In *WACV*, pages 2322–2331, 2021.

[Kendall and Gal, 2017] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NeurIPS*, 2017.

[Kendall *et al.*, 2015] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. In *BMVC*, 2015.

[Kingma and Welling, 2014] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[Lea *et al.*, 2016] Colin Lea, Austin Reiter, René Vidal, and Gregory D Hager. Segmental spatiotemporal cnns for fine-grained action segmentation. In *ECCV*, pages 36–52, 2016.

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

[Lea *et al.*, 2017b] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *CVPR*, pages 156–165, 2017.

[Lei and Todorovic, 2018] Peng Lei and Sinisa Todorovic. Temporal deformable residual networks for action segmentation in videos. In *CVPR*, pages 6742–6751, 2018.

[Li *et al.*, 2020] Shi-Jie Li, Yazan AbuFarha, Yun Liu, Ming-Ming Cheng, and Juergen Gall. Ms-tcn++: Multi-stage temporal convolutional network for action segmentation. *TPAMI*, pages 1–1, 2020.

[Li *et al.*, 2021] Shuang Li, Yilun Du, Antonio Torralba, Josef Sivic, and Bryan Russell. Weakly supervised human-object interaction detection in video via contrastive spatiotemporal regions. In *ICCV*, pages 1845–1855, 2021.

[MacKay, 1992] David JC MacKay. A practical bayesian framework for backpropagation networks. *NC*, 4(3):448–472, 1992.

[Menapace *et al.*, 2021] Willi Menapace, Stephane Lathuiliere, Sergey Tulyakov, Aliaksandr Siarohin, and Elisa Ricci. Playable video generation. In *CVPR*, pages 10061–10070, 2021.

[Sridhar *et al.*, 2021] Deepak Sridhar, Niamul Quader, Srikanth Muralidharan, Yaoxin Li, Peng Dai, and Juwei Lu. Class semantics-based attention for action detection. In *ICCV*, pages 13739–13748, 2021.

[Stein and McKenna, 2013] Sebastian Stein and Stephen J McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *ACM IJCPUC*, pages 729–738, 2013.

[Sultani *et al.*, 2018] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *CVPR*, pages 2942–2950, 2018.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NuerIPS*, pages 5998–6008, 2017.

[Wang *et al.*, 2020] Zhenzhi Wang, Ziteng Gao, Limin Wang, Zhifeng Li, and Gangshan Wu. Boundary-aware cascade networks for temporal action segmentation. In *ECCV*, pages 34–51, 2020.

[Yi *et al.*, 2021] Fangqiu Yi, Hongyu Wen, and Tingting Jiang. Asformer: Transformer for action segmentation. In *BMVC*, pages 1–15, 2021.