Differmer for Action Segmentation

Nicolas Aziere
Oregon State University
azieren@oregonstate.edu

Tieqiao Wang
Oregon State University
wangtie@oregonstate.edu

Sinisa Todorovic
Oregon State University
sinisa@oregonstate.edu

Abstract

We propose a novel approach to supervised action segmentation that explicitly models uncertainty over frame-wise class predictions using the Dirichlet distribution. In contrast to most SOTA methods that rely on the multi-stage refinement of initially proposed frame labels, our approach recalibrates frame-level class distributions through a Dirichlet diffusion process, which is analytically tractable (closed-form) and hence computationally efficient. Diffusion parameters are estimated only at a sparse set of keyframes using a lightweight module, further reducing memory and runtime costs. Experiments on four benchmark datasets – Breakfast, GTEA, 50Salads, and Assembly101 – show that our approach achieves superior accuracy with fewer parameters and lower computational complexity than existing approaches.

1. Introduction

This paper addresses supervised action segmentation in untrimmed videos—a basic vision problem—where every video frame is labeled with an action class. Recent work typically uses multi-stage architectures [5, 14, 33, 49], where frame labels predicted by the first stage are further refined by subsequent stages. Each stage aims to capture short- and long-range temporal dependencies for refining frame labels, but operates on individual videos without explicitly modeling a prior over prediction uncertainty.

Following prior work, we also aim to refine the initially proposed framewise classifications – but uniquely, we do so by incorporating a categorical prior distribution over the predicted class probabilities at each frame. For multi-class settings, where each frame's prediction is a discrete probability distribution across mutually exclusive action classes, the Dirichlet distribution serves as a natural choice for this prior, allowing us to capture epistemic uncertainty and guide the refinement process. Instead of relying on multi-stage refinement layers, we employ a diffusion process guided by the Dirichlet prior – referred to as Dirichlet diffusion [6] – to recalibrate frame-level class distributions. As we show, this diffusion process is analytically tractable with a closed-form solution, enabling efficient computation.

The Dirichlet diffusion process is controlled by two sets of learnable parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\kappa}$, as illustrated in Fig. 1. The figure presents toy examples showing how an initial 3-class probability distribution evolves over the simplex into a refined distribution under different values of $\boldsymbol{\alpha}$ and $\boldsymbol{\kappa}$. We estimate these parameters only at a sparse set of keyframes and then propagate them to the other frames for performing the Dirichlet diffusion process on all frames. For parameter estimation, we introduce a lightweight module, which is differentiable and well integrated, end-to-end, in our action segmentation model.

By eliminating multiple refinement stages used in SOTA methods, we reduce memory complexity. Further runtime reduction is achieved by estimating the Dirichlet diffusion parameters only at a sparse set of keyframes, rather than across all frames.

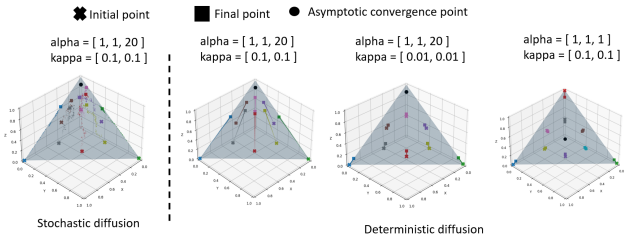


Figure 1. Example Dirichlet diffusion trajectories in 3D simplex space for different values of Dirichlet parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\kappa}$. Starting from an initial 3-class probability distribution (marked by a cross), the diffusion process converges to a destination distribution (marked by a square) determined by $\boldsymbol{\alpha}$'s , while $\boldsymbol{\kappa}$'s control the sharpness and directionality of the diffusion trajectory.

Although the first stage of our action segmentation model, responsible for proposing initial frame-level class distributions, can be implemented using any recent deep architecture, we adopt the first stage of ASFormer [49] and extend it with a lightweight module for Dirichlet parameter estimation and diffusion-based class distribution refine-

ment. This integration gives rise to our model, which we call Differmer.

Our experimental results demonstrate that Differmer outperforms SOTA models on the benchmark Breakfast [29], GTEA [15], 50Salads [43], and Assembly101 [39] datasets, while also reducing both the number of model parameters and overall time complexity.

In the following, Sec. 2 reviews related work, Sec. 3 describes the Dirichlet diffusion, Sec. 4 specifies Differmer, and Sec. 5 presents experimental results.

2. Related Work

Action segmentation of untrimmed videos has undergone tremendous progress with the recent advent of multi-stage deep models. For example, MS-TCN [14, 33] uses multiple stages of temporal convolutions over increasingly longer temporal windows. ASFormer [49] extends the Visual Transformer [13] to the video domain, and consists of an encoder and a sequence of decoder blocks. The encoder of ASFormer iteratively enriches frame features through self-attention over a pre-defined hierarchy of temporal windows. Each decoder block of ASFormer refines predictions of the encoder and previous decoder, and has a similar architecture as the encoder. TCTr [3] has a hybrid architecture, featuring convolution in the self-attention of a transformer. LTContext [5] also combines a transformer with temporal convolution to iterate between computing windowed local attention and sparse long-term context attention. Other models focus on feature enhancement [4, 19, 32, 47], boundary refinement [25, 48], or hierarchical reasoning over actions [1, 7, 18]. DiffAct [35] is the most closely related model to ours, since it also uses diffusion for action segmentation. DiffAct uses the *generative* Gaussian diffusion model of [22] for *denoising* a sequence of frame probabilities (a.k.a. the reverse process), conditioned on the frame features. In contrast, we resort to a *deterministic* Dirichlet diffusion process that is mathematically more suitable for modeling categorical class-assignment distributions of video frames. Importantly, our goal is not the reverse Gaussian denoising, but re-calibrating confidence of the initial prediction in the *forward* Dirichlet diffusion process. Our diffusion offers the closed-form solution resulting in a simpler and lighter model than the multistage refinement without needs for many iterative diffusion steps.

Aleatoric and epistemic uncertainty have been long studied in video understanding [2, 10, 11, 20, 21, 23, 31]. Aleatoric uncertainty [24] arises from inexplicable randomness of data, and is typically addressed with ensemble methods or Bayesian techniques [8, 30, 38]. Epistemic uncertainty [17, 24, 27, 37] stems from shortcomings of the model, training procedure, or inductive bias. Approaches to estimation of epistemic uncertainty include: variational Bayesian methods aimed at learning a distribution of model parameters [16], calibration methods aimed at estimating a distribution over prediction probabilities [19], and prior networks [26, 40, 44]. For action segmentation in [10], frame-wise uncertainty of features is estimated with Monte-Carlo sampling, and then used to refine action-class predictions for every frame. Closely related to our Differmer are approaches that use Dirichlet Prior Network (DPN) [38] and Evidence Neural Network (ENN) [40] for estimating parameters of the Dirichlet categorical distribution in order to quantify image classification uncertainty. One difference is that both DPN and ENN are convolutional whereas we use a transformer-based network. Another difference is that DPN requires external out-of-distribution (OOD) data samples in training. We adopt the ENN's more general approach to learning the Dirichlet parameters without using external OOD samples.

3. Review of the Dirichlet Diffusion Process

This section briefly reviews the underlying theory of the Dirichlet diffusion.

Let $\boldsymbol{p} = [p_1, \ldots, p_c, \ldots, p_C]^\top \in [0,1]^C$, $\sum_{c=1}^{C} p_c = 1$, denote an action-class distribution predicted for an input frame feature, $\boldsymbol{x} \in \mathbb{R}^d$. Instead of considering $\boldsymbol{p}$ as a point estimate, we model a categorical distribution of $\boldsymbol{p}$ with the following Dirichlet distribution:

$$D(\boldsymbol{p}; \boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{c=1}^{C} p_c^{\alpha_c - 1} \tag{1}$$

where B is the beta function [28], and parameters $\boldsymbol{\alpha} = \{\alpha_c : \alpha_c \geq 1,\ c = 1, \ldots, C\}$, denote strength of the corresponding probabilities in $\boldsymbol{p}$ over the C classes. As explained in Sec. 4, we estimate $\boldsymbol{\alpha}$ as a function of input frame features, $\boldsymbol{\alpha} = \boldsymbol{\alpha}(\boldsymbol{x})$, and in this way seek to estimate uncertainty of the prediction $\boldsymbol{p}$. This is used to re-calibrate the initial $\boldsymbol{p}$ by the Dirichlet diffusion.

The entropy of the Dirichlet distribution, $H(\boldsymbol{\alpha}; \boldsymbol{x})$, can be used to quantify uncertainty of prediction $\boldsymbol{p}$ for frame $\boldsymbol{x}$, where the lower the entropy the higher the confidence in prediction. The entropy $H(\boldsymbol{\alpha}; \boldsymbol{x})$ has the following closed form:

$$H(\boldsymbol{\alpha}; \boldsymbol{x}) = \log B(\boldsymbol{\alpha}) + (\alpha_0 - C)\psi(\alpha_0) - \sum_{c=1}^{C} (\alpha_c - 1)\psi(\alpha_c), \tag{2}$$

where each $\alpha_c = \alpha_c(\boldsymbol{x})$, $\alpha_0 = \sum_{c=1}^{C} \alpha_c(\boldsymbol{x})$, and $\psi(\cdot)$ is the digamma function [28].

Our approach uses the Dirichlet diffusion [6] to modify the initial class distribution $\boldsymbol{p}$ along a trajectory within the simplex space, as illustrated in Fig. 1. The trajectory follows a stochastic differential equation with the Dirichlet distribution, given by (1), as its asymptotic solution. The

update rule of the class distribution $\boldsymbol{p}(\tau)$ is defined as

$$p_c(\tau + d\tau) = p_c(\tau) + dp_c(\tau), \text{ for } c = 1, \dots, C-1 \ , \quad (3)$$

where the last logit $p_C = 1 - \sum_{c=1}^{C-1} p_c$. The offset $dp_c(\tau)$ in (3) can be estimated as a function of $\boldsymbol{\alpha}$ and $\boldsymbol{\kappa} = \{\kappa_c : \kappa_c \in (0,1), \ c = 1, \dots, C-1\}$ parameters, where the former govern the trajectory's asymptotic destination point at convergence, and the latter control the "speed" of moving along the trajectory. While we defer the derivation of $dp_c(\tau)$ to the supplemental material, the update rule in (3) can be expressed as

$$\boldsymbol{p}(\tau + d\tau) = \boldsymbol{p}(\tau) + M(\boldsymbol{\alpha}, \boldsymbol{\kappa})\, \boldsymbol{p}(\tau)\, d\tau + \boldsymbol{g}(\boldsymbol{\kappa}, \boldsymbol{p}(\tau))\, \eta\, d\tau \ , \quad (4)$$

where η represents the Gaussian noise, $\eta \sim \mathcal{N}(0,1)$, and the matrix $M \in \mathbb{R}^{C \times C}$ and vector $\boldsymbol{g} \in R^C$ are computed to ensure that $\sum_{c=1}^{C} p_c(\tau + d\tau) = 1$.

To reduce the computational cost of sampling multiple stochastic trajectories in (4), in this paper, we use the closed-form deterministic Dirichlet diffusion by estimating the expected trajectory after a finite number of steps n as:

$$\mathbb{E}\Big[\boldsymbol{p}(\tau + d\tau)\Big] = \boldsymbol{p}(\tau) + M(\boldsymbol{\alpha}, \boldsymbol{\kappa}) \boldsymbol{p}(\tau) d\tau \quad (5)$$

$$\Rightarrow \quad \boldsymbol{p}(n\, d\tau) = \Big(I + M(\boldsymbol{\alpha}, \boldsymbol{\kappa}) d\tau\Big)^n \boldsymbol{p}(0), \quad (6)$$

where I is the $C \times C$ identity matrix, and $n = 400$ and $d\tau = 10^{-4}$ are empirically found as optimal. From our experiments, the expected trajectory in (6) reduces randomness and thus better facilitates end-to-end learning of the Dirichlet parameters than the stochastic diffusion in (4).

4. Overview of Differmer

Fig. 2 shows an overview of Differmer which consists of three modules.

Our first module is equivalent to the encoder of ASFormer [49]. Given video frame features at the input, $\mathcal{X} = \{\boldsymbol{x}_t : \boldsymbol{x}_t \in \mathbb{R}^D, \ t = 1, \dots, T\}$, the encoder provides framewise softmax predictions over the set of action classes, $\mathcal{C}$, $|\mathcal{C}| = C$, and deep features enriched with self-attention over temporal windows with increasing sizes, $\{(\boldsymbol{p}_t, \boldsymbol{f}_t) : \boldsymbol{p}_t \in [0,1]^C, \ \boldsymbol{f}_t \in \mathbb{R}^d, \ t = 1, \dots, T\}$. The framewise predictions of the first stage can be mapped to the MAP framewise classification, $\hat{\mathcal{Y}} = \{\hat{y}_t : \hat{y}_t = \arg\max_{c \in \mathcal{C}} \boldsymbol{p}_t, \ t = 1, \dots, T\}$, resulting in a transcript, i.e., a sequence of predicted action segments, $\hat{\mathcal{S}} = \{(\hat{y}_s, \hat{l}_s) : \hat{y}_s \in \mathcal{C}, \ l_s \in \mathbb{N}_+, \ s = 1, \dots, S\}$, where $\hat{l}_s$ denotes the predicted length of an action segment in the video. Tab. 1 shows that the initial prediction $\hat{\mathcal{S}}$ from the first stage of ASFormer achieves a high true positive rate in detecting action boundaries on the 50Salads dataset [43].

The next module of Differmer estimates Dirichlet parameters to enable diffusion-based refinement of the predicted framewise class distributions $\{\boldsymbol{p}_t : t = 1, \dots, T\}$.

Model	MSTCN [33]	LTContext [5]	ASFormer [49]
TPR@10	45.2	40.2	40.7
TPR@50	77.4	78.9	84.9

Table 1. True positive rates (TPR) of action-boundary detection by the first stage of SOTA models within 10- and 50-frame windows around ground-truth boundaries on the 50Salads dataset [43].

Since we empirically find that $\hat{\mathcal{S}}$ provides reliable action-boundary detections (see Tab. 1), we improve efficiency by selecting a single keyframe $k \in [T_s^{\text{start}}, T_s^{\text{end}}]$ from each predicted action segment $s \in \hat{\mathcal{S}}$. Dirichlet parameters are then estimated only at these keyframes, reducing computational cost. The keyframe is chosen as the frame with the highest classification score for the predicted class of s: $k = \arg\max_{t \in [T_s^{\text{start}}, T_s^{\text{end}}]} p_{\hat{y}_s, t}$.

After selecting the keyframes, their softmax predictions and deep features, $\{(\boldsymbol{p}_k, \boldsymbol{f}_k) : k = 1, ..., S\}$, are passed to a two-layer feedforward MLP followed by the standard Multi-Head-Self-Attention (MHSA) [45], and then input to the Alpha and Kappa networks to predict the corresponding Dirichlet parameters $\{\boldsymbol{\alpha}_k\}$ and $\{\boldsymbol{\kappa}_k\}$. The Alpha and Kappa networks have the same architecture, but do not share the weights, and consist of one MHSA with a global receptive field and a one-layer MLP for predicting their respective outputs. In the one-layer MLP, we use the standard sigmoid activation for predicting $\boldsymbol{\kappa}$. For $\boldsymbol{\alpha}$, we compute the following strictly positive activation function:

$$\boldsymbol{\alpha}_k(\boldsymbol{f}_k|\theta) = \mathbf{1} + \exp(\text{AlphaNet}(\boldsymbol{f}_k|\theta)), \quad k = 1, \dots, S \quad (7)$$

where $\mathbf{1}$ is the C-dimensional vector of 1's, and θ are the network parameters.

For every keyframe k, the estimated $\boldsymbol{\alpha}_k$ and $\boldsymbol{\kappa}_k$ are used in the Dirichlet diffusion process to efficienlty refine the softmax predictions $\boldsymbol{p}_k$, using the closed-form expression of the Dirichlet diffusion trajectory given by (6). Finally, we upsample the refined class distributions to all frames as follows. All video frames belonging to the same action segments as their keyframes inherit the corresponding diffused $\{\boldsymbol{p}_k\}$, and the estimates $\{\boldsymbol{\alpha}_k\}$ and $\{\boldsymbol{\kappa}_k\}$, resulting in the framewise outputs $\{(\boldsymbol{p}_t, \boldsymbol{\alpha}_t, \boldsymbol{\kappa}_t) : t = 1, \dots, T\}$. The diffused and upsampled $\{\boldsymbol{p}_t\}$ are used to update the MAP framewise classification $\hat{\mathcal{Y}}$. This enables re-selection of new keyframes to pass to the new block for re-estimation of $\{\boldsymbol{\alpha}_k\}$ and $\{\boldsymbol{\kappa}_k\}$, and re-diffusion of $\{\boldsymbol{p}_k\}$.

The framewise outputs incur loss $\mathcal{L}$ for our end-to-end training of Differmer. $\mathcal{L}$ includes the cross-entropy loss, alpha loss, and kappa loss:

$$\mathcal{L} = \frac{1}{T} \sum_{t=1}^{T} \Big[\mathcal{L}_{\text{CE}}(\boldsymbol{p}_t, \boldsymbol{y}_t) + \lambda_\alpha \mathcal{L}_\alpha(\boldsymbol{\alpha}_t, \boldsymbol{y}_t) + \lambda_\kappa \mathcal{L}_\kappa(\boldsymbol{\kappa}_t, \boldsymbol{\alpha}_t, \boldsymbol{y}_t)\Big], \quad (8)$$

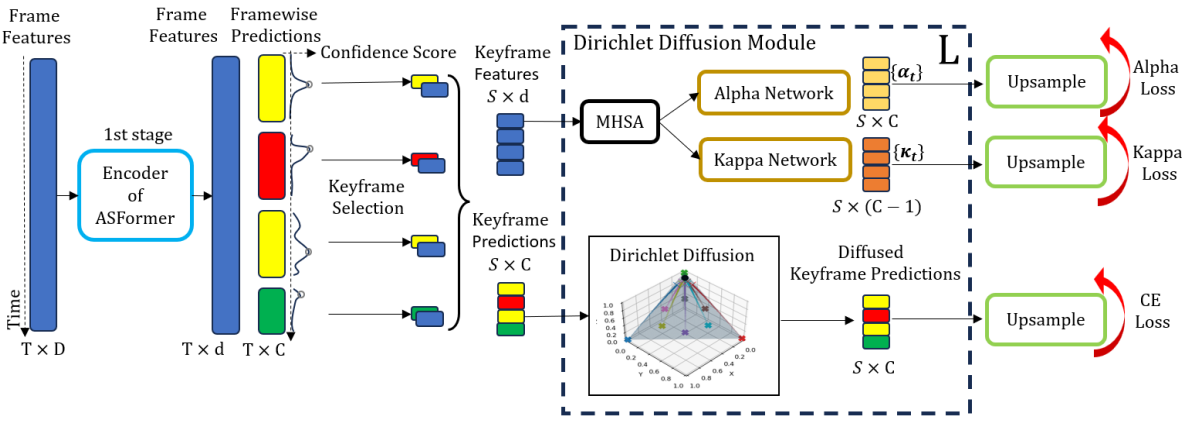


Figure 2. Differmer: The encoder of ASFormer [49] provides framewise class predictions. Features and predictions of keyframes, each representing the respective action segment of the initial prediction, are passed to the diffusion module with L consecutive blocks ($L = 2$). In each block, there are Multi-Head Self-Attention (MHSA) and two transformer-based networks that estimate the Dirichlet parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\kappa}$ for diffusing the initial class predictions of the keyframes. Video frames belonging to the same action segments as their keyframes inherit the corresponding $\boldsymbol{\alpha}$'s, $\boldsymbol{\kappa}$'s, and diffused predictions. This incurs the cross-entropy (CE) loss, alpha loss, and kappa loss.

where $\boldsymbol{y} = [y_1, \ldots, y_c, \ldots, y_C] \in \{0, 1\}^C$ denotes the one-hot ground-truth vector, and λ_α and λ_κ are positive hyper-parameters. Loss $\mathcal{L}$ supervises each block of the Diffusion module. Below, we specify the alpha loss and kappa loss.

With $\mathcal{L}_\alpha$, our goal is to learn the Dirichlet parameters $\boldsymbol{\alpha}$ which minimize the Bayes risk of the class predictor, as in [40]:

$$\mathcal{L}_\alpha(\boldsymbol{\alpha}, \boldsymbol{y}) = \int \Big[\sum_{c=1}^{C} -y_c \log p_c \Big] \frac{1}{B(\boldsymbol{\alpha})} \prod_{c=1}^{C} p_c^{\alpha_c - 1} d\boldsymbol{p} \quad (9)$$

$$\approx \sum_{c=1}^{C} y_c \Big(\psi(\alpha_0) - \psi(\alpha_c) \Big), \quad (10)$$

where in (10), we use the Dirichlet distribution $D(\boldsymbol{p}; \boldsymbol{\alpha})$, given by (1), as the prior of prediction $\boldsymbol{p}$. By minimizing $\mathcal{L}_\alpha(\boldsymbol{\alpha}, \boldsymbol{y})$ in (10), we enforce that the alpha for the ground-truth class is greater than the sum of the other alpha's.

To specify $\mathcal{L}_\kappa$, recall that $\boldsymbol{\kappa}$ controls the "speed" of the Dirichlet diffusion to the asymptotic point, and that the asymptotic point depends on $\boldsymbol{\alpha}$. The larger the sum of all kappa's, the stronger the diffusion updates. It seems reasonable to enforce the diffusion trajectory to take longer strides toward the destination, i.e., to maximize the sum of kappa's, when α_c for the ground-truth class $y_c = 1$ is the largest value in $\boldsymbol{\alpha}$. Conversely, if the asymptotic point of the diffusion trajectory disagrees with the ground truth, we would like to minimize modification of the initial prediction $\boldsymbol{p}$ by the diffusion, i.e., to minimize the sum of kappa's. Thus, we formulate $\mathcal{L}_\kappa$ as

$$\mathcal{L}_\kappa(\boldsymbol{\kappa}, \boldsymbol{\alpha}, \boldsymbol{y}) = -b(\boldsymbol{y}, \boldsymbol{\alpha}) \left(\sum_{c=1}^{C-1} \log \kappa_c \right), \quad (11)$$

where $b(\boldsymbol{y}, \boldsymbol{\alpha})$ is a binary function:

$$b(\boldsymbol{y}, \boldsymbol{\alpha}) = \begin{cases} 1 & , \text{ if } \alpha_c = \max(\boldsymbol{\alpha}) \text{ and } y_c = 1, \\ -1 & , \text{ otherwise.} \end{cases} \quad (12)$$

5. Results

Datasets include four benchmarks: Breakfast [29], GTEA [15], 50Salads [43] and Assembly101 [39]. *Breakfast* has 1712 videos with 48 action classes, where the video lengths vary from 30 seconds (e.g. for "cereals") to 5 minutes (e.g. for "pancakes"). *GTEA* has 28 videos showing 7 complex activities performed by 4 different people; each activity is specified in terms of 11 action classes including the background class. *50Salads* consists of 50 videos with 17 action classes. *Assembly101* is the most challenging dataset, and consists of 4136 videos with 202 action classes. For Assembly101, the videos are about 6-12 minutes long; the same action class may have many instances in a video, and transcripts have on average 24 actions in the sequence. For the Breakfast, GTEA, and 50Salads datasets, we pre-computed MAEv2 frame features using the SOTA action recognition model VideoMAEv2 [46], in the same way the I3D features [9] were extracted in [14]. For Breakfast, GTEA, and 50Salads, we perform the standard 4-fold, 4-fold, and 5-fold cross validation, respectively, on MAEv2 features. Assembly101 has only one split, and for our evaluation we use its original TSM features [34].

Metrics. Mean-of-Frame (MoF) is the average framewise classification accuracy. Edit score counts edit operations to make the predicted and ground-truth sequences equivalent. F1 score counts true positives when a tem-

poral intersection between the predicted and ground-truth action segments is 10%, 25% and 50%, denoted as F1@{10,25,50}.

Implementation details. Differmer's first module has the same architecture as the encoder of ASFormer [49]. For each dataset, Differmer was trained for 120 epochs on an NVIDIA GeForce GTX 1080 GPU, with a mini-batch of size 1. The Alpha and Kappa networks consist of one multi-head self-attention layer and one 1D-Convolution layer with 8 heads. Regarding the Dirichlet Diffusion process parameters, we use $d\tau = 10^{-4}$ and $n = 400$ for all datasets. We set $\lambda_\alpha = 0.8$ and $\lambda_\kappa = 0.1$ for all datasets.

5.1. Ablation Study

The ablation study tests our individual contributions on 50Salads, as the standard dataset for evaluating ablations in recent work [5, 49]. All results of the ablation tests are obtained for MAEv2 frame features.

Number of stages. Tab. 2 shows how different numbers of the consecutive diffusion blocks in the Diffusion module of Differmer affect its performance. With only 1 diffusion block, Differmer with a total of 0.44M parameters gives results that come very close to those of the full ASFormer with 1.09M parameters. As the number of the diffusion blocks increases above 3, we observe marginal performance gains. Therefore, as a good trade off between model complexity and performance, in the remaining experiments we use the Diffusion module with 2 diffusion blocks.

# of Diffusion Blocks	F1@{10,25,50}			Edit	MoF	# of Parameters (M)
0	62.6	61.5	57.9	55.4	89.4	0.38
1	91.4	90.8	84.0	86.7	89.7	0.44
2	**91.8**	**90.9**	**87.5**	**87.0**	**91.5**	**0.48**
3	91.4	90.7	87.1	86.7	89.6	0.49

Table 2. Impact of the number of the diffusion blocks in Differmer on our performance on 50salads. When there are no diffusion blocks (0), Differmer is actually the encoder of ASFormer [49]. 2 blocks give a good trade off between complexity and performance.

Number of diffusion iterations. In Tab. 3, we vary the number of diffusion steps n to update class predictions for the keyframes, given by (6). Using α directly as output is represented by the row with ∞ symbol, it is equivalent to when the diffusion converges toward the mean of the Dirichlet distribution. In the following, we use $n = 400$ steps when Differmer achieves the best performance.

Backbone Choice. Tab. 4 presents a comparison of true positive rates (TPR) and false positive rates (FPR) for action-boundary detection, evaluated within temporal windows of 5, 10, and 50 frames around the ground-truth boundaries. This comparison focuses on three state-of-the-art (SOTA) multi-stage models after their first and third stages. Our decision to use the encoder of ASformer as the backbone network is based on its superior TPR@50 performance among the three models after the first stage. A high TPR combined with a low FPR is crucial in ensuring accurate boundary detection while minimizing false alarms. Our design choice to use the encoder of ASformer as our backbone network is informed by its superior TPR@50 vs FPR@50 trade-off among the three models after 1st stage.

n	F1@{10,25,50}			Edit	MoF
100	90.3	89.0	86.5	85.0	89.8
200	90.6	89.0	86.1	85.7	89.7
400	**91.8**	**90.9**	**87.5**	**87.0**	**91.5**
800	91.2	90.6	87.0	86.6	90.1
∞	89.5	88.1	85.1	83.8	88.8

Table 3. Differmer's performance on 50Salads for different numbers of the Dirichlet diffusion steps n. Differmer achieves the best performance for $n = 400$. ∞ means the prediction is done using only α without any diffusion steps.

Differmer shows an improvement in TPR@50, achieving 54.4 with one diffusion block and increasing to 59.7 with two diffusion blocks, while its FPR@5 decreases from 28.3 to 32.0. This consistent performance across all TPR metrics and reductions in FPR metrics indicate the effectiveness of Differmer as it progresses from its first stage (the encoder of ASformer) to the first diffusion block, and from the first to the second diffusion block.

Model	TPR@5	TPR@10	TPR@50	FPR@5	FPR@10	FPR@50
MSTCN [33] (1 stage)	31.2	45.2	77.4	95.9	94.1	89.9
MSTCN [33] (3 stages)	17.1	29.1	70.3	84.0	72.8	34.3
ASformer [49] (1 stage)	22.1	40.7	84.9	72.8	60,7	79.0
ASformer [49] (3 stages)	31.2	42.7	75.9	85.9	78.8	28.4
LTContext [5] (1 stage)	24.6	40.2	78.9	89.0	82.0	64.6
LTContext [5] (3 stages)	18.6	30.1	74.4	82.2	71.1	28.8
Differmer (1 diff. block)	28.3	44.9	86.3	71.2	54.4	62.1
Differmer (2 diff. blocks)	32.0	43.4	77.0	74.4	59.7	25.0

Table 4. True and false positive rates (TPR, FPR) of action-boundary detection on 50salads after a particular stage of a given model.

Differmer = Backbone + Diffusion module	F1@{10,25,50}			Edit	MoF
1st stage of MSTCN [14] + Diffusion module	86.8	85.1	79.5	79.7	89.0
1st stage of LTContext [5] + Diffusion module	90.4	89.8	83.8	85.0	89.3
encoder of ASFormer [49] + Diffusion module	**91.8**	**90.9**	**87.5**	**87.0**	**91.5**

Table 5. Differmer's performance on 50Salads for different backbone networks selected as the first stage of Differmer. As the backbone, we use the first stage of the respective SOTA models (not full models).

Tab. 5 evaluates performance contributions of different backbone networks taken as the first stage of Differmer. Using the encoder of ASFormer [49] as our 1st stage gives the best action segmentation on 50Salads.

Correction of the initial uncertainty. Fig. 3 is an Expected Calibration Error Plot where we shows that the Dirichlet diffusion corrects confidence in the initial frame-wise prediction made by the first stage of Differmer. Confidence is estimated as the entropy of the Dirichlet distribution of the predictions, given by (2), where the lower the en-

tropy the higher the confidence. In the figure, the x-axis normalizes the entropy values of all frames from all test videos in 50Salads into bins, from the lowest 0-5% to highest 95-100%. The y-axis of the plot evaluates classification accuracy of the frames whose Dirichlet entropy falls in a particular bin. As we can see, for the bin 0-5%, the number of incorrectly classified frames with the highest confidence in prediction reduces after the diffusion. This suggests that the Dirichlet diffusion corrects uncertainty after the first stage such that the framewise class predictor is more uncertain for incorrectly classified frames, as desirable.

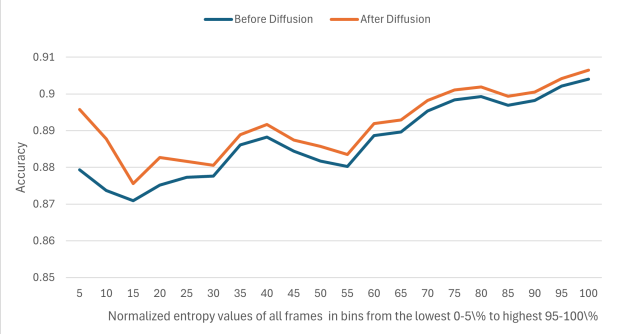


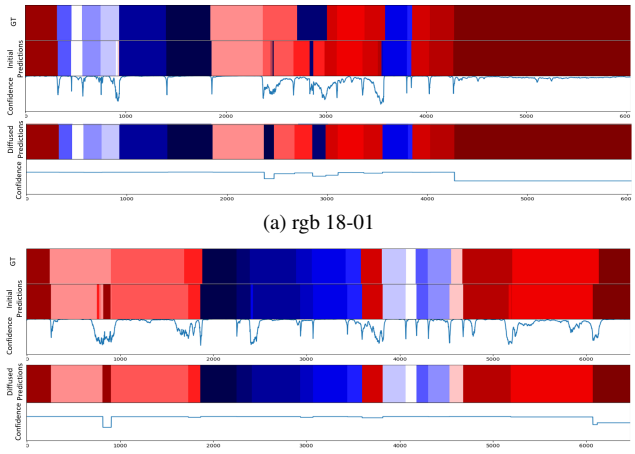
Figure 3. Expected Calibration Error with Entropy. Classification accuracy of all video frames from 50Salads whose Dirichlet entropy, given by (2), falls in a particular bin along the x-axis. The bins normalize the entropy values from the lowest 0-5% to highest 95-100%. For the bin 0-5%, the number of incorrectly classified frames with the highest confidence in prediction reduces after the diffusion. This suggests that the diffusion re-calibrates the initial uncertainty.

Loss functions. Tab. 6 presents the effect of $\mathcal{L}_\alpha$, given by (10), and $\mathcal{L}_\kappa$, given by (11), on Differmer's performance. When the two loss functions are not used, the supervision for learning the Alpha network and the Kappa network comes only from the cross-entropy loss in (8). From Tab. 6, regularization of learning $\boldsymbol{\alpha}$ and $\boldsymbol{\kappa}$ with $\mathcal{L}_\alpha$ and $\mathcal{L}_\kappa$ improves results, and the best performance is achieved when both $\mathcal{L}_\alpha$ and $\mathcal{L}_\kappa$ are added to the cross entropy loss.

$\mathcal{L}_\alpha$	$\mathcal{L}_\kappa$	F1@{10,25,50}			Edit	MoF
no	no	89.1	88.4	84.1	83.5	89.0
✓	no	90.3	89.8	85.1	85.5	90.7
no	✓	89.5	89.0	85.7	84.5	88.9
✓	✓	**91.8**	**90.9**	**87.5**	**87.0**	**91.5**

Table 6. Regularization of learning $\boldsymbol{\alpha}$ and $\boldsymbol{\kappa}$ with $\mathcal{L}_\alpha$ and $\mathcal{L}_\kappa$ improves results of Differmer on 50salads. When $\mathcal{L}_\alpha$ and $\mathcal{L}_\kappa$ are not used, the supervision for learning the Alpha network and the Kappa network comes only from $\mathcal{L}_{\text{CE}}$ in (8).

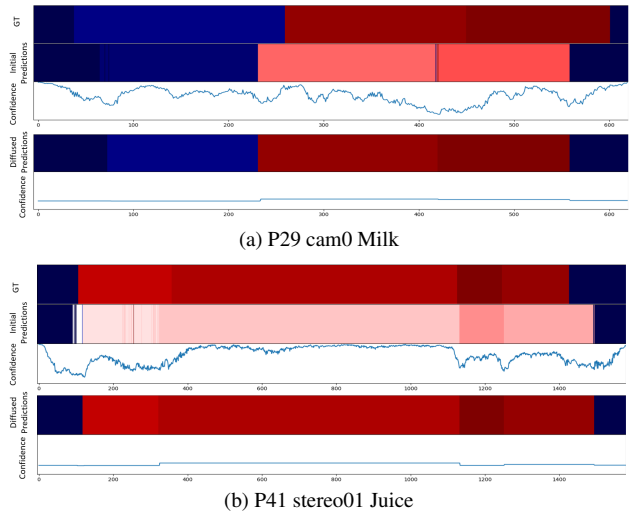
I3D vs. MAEv2 frame features. Tab. 7 shows that using MAEv2 frame features as input to the SOTA models and Differmer, across the board, gives better action segmentation than I3D features [9]. MAEv2 frame features are extracted with VideoMAEv2 model [46] trained on the Kinetic-700 dataset [42]. For fair comparison, we retrained and evaluated MS-TCN [14], ASFormer [49], and LTContext [5] with MAEv2 features.



(a) rgb 18-01

(b) rgb 17-02

Figure 4. Action segmentation on two example videos from 50salads (best viewed in color). (top row) Ground Truth; (2nd row from the top) First-stage prediction by the encoder of ASFormer in Differmer; (3rd row from the top) First-stage confidence scores for every frame; (4th row) Diffused prediction; (bottom row) Diffused confidence scores for every frame.



(a) P29 cam0 Milk

(b) P41 stereo01 Juice

Figure 5. Action segmentation on two example videos from Breakfast (best viewed in color). (top row) Ground Truth; (2nd row from the top) First-stage prediction by the encoder of ASFormer in Differmer; (3rd row from the top) First-stage confidence scores for every frame; (4th row) Diffused prediction; (bottom row) Diffused confidence scores for every frame.

Dataset	50Salads					GTEA					Breakfast				
Method	F1@{10,25,50}			Edit	MoF	F1@{10,25,50}			Edit	MoF	F1@{10,25,50}			Edit	MoF
MSTCN [14]	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
MSTCN++ [33]	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
C2F-TCN [41]	75.6	72.7	61.2	69.1	79.6	89.9	88.3	75.9	86.8	79.6	64.9	60.6	49.7	63.2	70.2
BCN [48]	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 [25]	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	90.9	88.6	76.4	87.5	78.7	74.7	69.5	57.0	71.9	69.4
SSTDA [12]	83.0	81.5	73.8	75.8	83.2	90.0	89.1	78.0	86.2	79.8	75.0	69.1	55.2	73.7	70.2
G2L [18]	80.3	78.0	69.8	73.4	82.2	89.9	87.3	75.8	84.6	78.5	74.9	69.0	55.2	73.3	70.7
UVAST [7]	81.2	70.4	83.9	77.1	69.7	92.7	91.3	81.0	**92.1**	80.2	76.7	70.0	56.6	68.2	68.2
ASFormer [49]	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
TCTr [3]	87.5	86.1	80.2	83.4	86.6	**93.7**	**92.4**	84.0	91.3	81.3	76.6	71.1	58.5	76.1	**77.5**
FACT [36]	85.5	83.4	77.4	79.2	86.8	93.5	92.1	84.1	91.4	**86.1**	**81.4**	**76.5**	**66.2**	**79.7**	76.2
LTContext [5]	89.4	87.7	82.0	83.2	87.7	-	-	-	-	-	77.6	72.6	60.1	77.0	74.2
DiffAct [35]	*90.1*	*89.2*	*83.7*	*85.0*	*88.9*	92.5	91.5	*84.7*	89.6	82.2	80.3	75.9	64.6	78.4	76.4
Differmer (I3D)	**90.3**	**89.4**	**85.3**	**85.2**	**89.2**	*93.1*	*91.8*	**85.0**	*90.5*	*82.3*	*80.7*	*76.1*	*64.8*	*78.4*	*77.2*
Bridge-Prompt [32]	89.2	87.8	81.3	83.8	88.1	94.1	92.0	83.0	91.6	81.2	-	-	-	-	-
MSTCN (MAEv2)	85.6	84.0	78.1	78.7	87.7	92.7	91.4	85.0	89.9	81.8	62.1	57.2	46.1	67.2	69.5
ASFormer (MAEv2)	89.7	88.6	84.5	84.0	89.9	93.4	92.6	85.1	90.3	82.1	77.2	72.0	59.2	75.6	75.8
LTContext (MAEv2)	89.5	87.9	83.1	83.9	89.1	-	-	-	-	-	80.8	75.9	63.1	76.0	76.1
FACT (MAEv2)	89.9	88.5	84.7	84.5	*89.3*	**94.7**	*93.7*	*86.4*	**93.4**	**86.7**	*80.9*	*76.3*	**65.5**	**79.4**	**77.8**
DiffAct (MAEv2)	*92.2*	*90.8*	*86.5*	*87.3*	89.2	93.8	93.2	87.5	91.9	82.4	80.3	76.1	64.9	78.6	77.2
Differmer (MAEv2)	**92.3**	**90.9**	**87.5**	**87.0**	**91.5**	*94.5*	**93.9**	**87.9**	*92.0*	*82.8*	**81.2**	**76.6**	*65.1*	*78.9*	*77.7*

Table 7. Comparison of Differmer with the SOTA models on 50Salads, GTEA and Breakfast, when using the I3D (top) and MAEv2 (bottom) frame features as input. The best results are colored bold black, and the second best are italic.

5.2. Qualitative Results

Fig. 4 and Fig. 5 visualize our results on example videos from 50Salads and Breakfast, respectively. Specifically, the figures show the confidence scores associated with every frame, and compare the action segmentation results before and after the Diffusion module. We observe that in most cases, the ground-truth boundaries are correctly estimated at our first stage by the encoder of ASFormer, while the confidence scores get lower at transitory moments between consecutive action instances in the video. From the figures, our diffusion process corrects the initial confidence scores, and for some segments switches their initially assigned class. While this re-labeling of segments often improves the initial prediction, in a few cases the initially correctly predicted class might be switched to a wrong class. Importantly, for wrongly classified segments, Differmer correctly estimates high uncertainty, i.e., a low confidence score.

5.3. Comparison with SOTA

Tab. 7 compares Differmer with the SOTA action segmenters, reviewed in Sec. 2, on the 50Salads, GTEA, and Breakfast datasets. The table is divided into three sections based on the type of frame features used as input: the top section reports results for I3D features, the middle for Bridge-Prompt features, and the bottom for MAEv2 features. With I3D features, Differmer outperforms most competing methods, second only to FACT [36], while offering significantly lower complexity, as highlighted in Tab. 9. When using MAEv2 features, Differmer closes the gap and achieves notable improvements across all datasets and metrics. Furthermore, Differmer consistently outperforms DiffAct, demonstrating the effectiveness of Dirichlet diffusion over Gaussian diffusion for refining categorical predictions in action segmentation.

Tab. 8 and Tab. 9 compare the performance and complexity of Differmer with those of the SOTA models. It is worth noting that this comparison is constrained to using the results reported in the literature on different datasets and different frame features – hence, splitting the comparison of complexity into two tables. Specifically, Tab. 8 compares Differmer and the SOTA action segmenters that reported their results and complexity on Assembly101 for the TSM frame features at the input. Tab. 9 compares Differmer and the SOTA models that reported their results and complexity on 50Salads for the I3D frame features at the input. Differmer has the least amount of parameters and achieves the best performance on both datasets.

Method	F1@{10,25,50}			Edit	MoF	# of Parameters (M)
MSTCN++ [33]	31.6	27.8	20.6	30.7	37.1	1.08
UVAST [7]	32.1	28.3	20.8	31.5	37.4	1.22
ASFormer [49]	33.4	29.2	21.4	30.5	38.8	1.13
LTContext [5]	**33.9**	**30.0**	**22.6**	**30.4**	**41.2**	**0.72**
Differmer	**34.6**	**30.8**	**23.5**	**31.3**	**42.2**	**0.60**

Table 8. Comparison of Differmer with the SOTA models on Assembly101, when using the TSM frame features as input. Differmer has the least amount of parameters and achieves the best performance. The best results are colored bold red, and the second best are colored bold black.

Method	MoF	# Parameters	FLOPS
ASFormer [49]	79.7	1.09M	5.77G
DiffAct [35]	88.9	0.975M	9.40G
FACT [36]	86.8	1.9M	5.5G
LTContext [5]	87.7	0.66M	8.56G
Differmer (I3D)	89.2	0.48M	2.45G

Table 9. Comparison of Differmer with the SOTA models on 50Salads, for the I3D frame features. Differmer has the least amount of parameters, smallest number of average FLOPs, the shortest average inference time, and achieves the best MoF.

6. Conclusion

We have proposed Differmer – a new model for action segmentation that refines the initial prediction of frame classes made by the first stage of a SOTA model using Dirichlet diffusion. Instead of treating these predictions as point estimates, we diffuse the framewise class distributions using their Dirichlet prior in the categorical space. For efficiency, the parameters of the Dirichlet prior are estimated only at keyframes using two transformer-based networks, instead of all frames. Differmer significantly reduces model complexity relative to the SOTA approaches which require multiple stages for refining the initial prediction. Moreover, due to our deterministic, closed-form Dirichlet diffusion and design choice to diffuse only keyframes, Differmer has a significantly lower computational complexity than SOTA. Our ablation study on 50Salads demonstrates that Differmer successfully increases true positive rate and decreases false positive rate of action boundary detection after the initial prediction. A comparison with SOTA on four benchmark datasets, including the largest Assembly101, indicates that Differmer achieves the best performance with the least complexity.

References

[1] Hyemin Ahn and Dongheui Lee. Refining action segmentation with hierarchical video representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16302–16310, 2021. 2, 7

[2] Murat Seckin Ayhan and Philipp Berens. Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks. *Medical Imaging with Deep Learning*, 2018. 2

[3] Nicolas Aziere and Sinisa Todorovic. Multistage temporal convolution transformer for action segmentation. *Image and Vision Computing*, 128:104567, 2022. 2, 7

[4] Nicolas Aziere and Sinisa Todorovic. Markov game video augmentation for action segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13505–13514, 2023. 2

[5] Emad Bahrami, Gianpiero Francesca, and Juergen Gall. How much temporal long-term context is needed for action segmentation? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10351–10361, 2023. 1, 2, 3, 5, 6, 7, 8

[6] J Bakosi, JR Ristorcelli, et al. A stochastic diffusion process for the dirichlet distribution. *International Journal of Stochastic Analysis*, 2013:1–7, 2013. 1, 2

[7] Nadine Behrmann, S Alireza Golestaneh, Zico Kolter, Juergen Gall, and Mehdi Noroozi. Unified fully and timestamp supervised temporal action segmentation via sequence to sequence translation. *arXiv preprint arXiv:2209.00638*, 2022. 2, 7

[8] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015. 2

[9] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 4, 6

[10] Lei Chen, Muheng Li, Yueqi Duan, Jie Zhou, and Jiwen Lu. Uncertainty-aware representation learning for action segmentation. *IJCAI*, 2022. 2

[11] Lei Chen, Muheng Li, Yueqi Duan, Jie Zhou, and Jiwen Lu. Uncertainty-aware representation learning for action segmentation. In *IJCAI*, 2022. 2

[12] Min-Hung Chen, Baopu Li, Yingze Bao, Ghassan AlRegib, and Zsolt Kira. Action segmentation with joint self-supervised temporal domain adaptation. In *CVPR*, 2020. 7

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv:2010.11929*, 2020. 2

[14] Yazan Abu Farha and Jurgen Gall. MS-TCN: Multi-stage temporal convolutional network for action segmentation. In *CVPR*, 2019. 1, 2, 4, 5, 6, 7

[15] Alireza Fathi, Xiaofeng Ren, and James M Rehg. Learning to recognize objects in egocentric activities. In *CVPR 2011*, pages 3281–3288. IEEE, 2011. 2, 4

[16] Yarin Gal and Zoubin Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*, 2015. 2

[17] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016. 2

[18] 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*, 2021. 2, 7

[19] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017. 2

[20] Hongji Guo, Zhou Ren, Yi Wu, Gang Hua, and Qiang Ji. Uncertainty-based spatial-temporal attention for online action detection. In *ECCV*, 2022. 2

[21] Hongji Guo, Hanjing Wang, and Qiang Ji. Uncertainty-guided probabilistic transformer for complex action recognition. In *CVPR*, 2022. 2

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[23] Po-Yu Huang, Wan-Ting Hsu, Chun-Yueh Chiu, Ting-Fan Wu, and Min Sun. Efficient uncertainty estimation for semantic segmentation in videos. In *ECCV*, 2018. 2

[24] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021. 2

[25] Yuchi Ishikawa, Seito Kasai, Yoshimitsu Aoki, and Hirokatsu Kataoka. Alleviating over-segmentation errors by detecting action boundaries. In *WACV*, 2021. 2, 7

[26] Taejong Joo, Uijung Chung, and Min-Gwan Seo. Being bayesian about categorical probability. In *International Conference on Machine Learning*, pages 4950–4961. PMLR, 2020. 2

[27] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015. 2

[28] Samuel Kotz, Narayanaswamy Balakrishnan, and Norman L Johnson. *Continuous multivariate distributions, Volume 1: Models and applications*. John Wiley & Sons, 2004. 2

[29] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *CVPR*, 2014. 2, 4

[30] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017. 2

[31] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*, 2017. 2

[32] Muheng Li, Lei Chen, Yueqi Duan, Zhilan Hu, Jianjiang Feng, Jie Zhou, and Jiwen Lu. Bridge-prompt: Towards ordinal action understanding in instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19880–19889, 2022. 2, 7

[33] Shi-Jie Li, Yazan AbuFarha, Yun Liu, Ming-Ming Cheng, and Juergen Gall. MS-TCN++: Multi-stage temporal convolutional network for action segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1, 2, 3, 5, 7

[34] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7083–7093, 2019. 4

[35] Daochang Liu, Qiyue Li, Anh-Dung Dinh, Tingting Jiang, Mubarak Shah, and Chang Xu. Diffusion action segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10139–10149, 2023. 2, 7, 8

[36] Zijia Lu and Ehsan Elhamifar. Fact: Frame-action cross-attention temporal modeling for efficient action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18175–18185, 2024. 7, 8

[37] David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992. 2

[38] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. *Advances in neural information processing systems*, 31, 2018. 2

[39] Fadime Sener, Dibyadip Chatterjee, Daniel Shelepov, Kun He, Dipika Singhania, Robert Wang, and Angela Yao. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21096–21106, 2022. 2, 4

[40] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31, 2018. 2, 4

[41] Dipika Singhania, Rahul Rahaman, and Angela Yao. C2f-tcn: A framework for semi-and fully-supervised temporal action segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):11484–11501, 2023. 7

[42] Lucas Smaira, João Carreira, Eric Noland, Ellen Clancy, Amy Wu, and Andrew Zisserman. A short note on the kinetics-700-2020 human action dataset. *arXiv preprint arXiv:2010.10864*, 2020. 6

[43] Sebastian Stein and Stephen J McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 729–738, 2013. 2, 3, 4

[44] Theodoros Tsiligkaridis. Information robust dirichlet networks for predictive uncertainty estimation, 2021. US Patent App. 17/064,046. 2

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 3

[46] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14549–14560, 2023. 4, 6

[47] Tieqiao Wang and Sinisa Todorovic. End-to-end action segmentation transformer. *arXiv preprint arXiv:2503.06316*, 2025. 2

[48] Zhenzhi Wang, Ziteng Gao, Limin Wang, Zhifeng Li, and Gangshan Wu. Boundary-aware cascade networks for temporal action segmentation. In *ECCV*, 2020. 2, 7

[49] Fangqiu Yi, Hongyu Wen, and Tingting Jiang. ASFormer: Transformer for action segmentation. *CoRR*, arXiv 2110.08568, 2021. 1, 2, 3, 4, 5, 6, 7, 8