Bayesian Evidential Deep Learning for Online Action Detection

Hongji Guo, Hanjing Wang, and Qiang Ji

Rensselaer Polytechnic Institute, Troy NY 12180, USA
{guoh11, wangh36, jiq}@rpi.edu

Abstract. Online action detection aims at identifying the ongoing action in a streaming video without seeing the future. Timely and reliable response is critical for real-world applications. In this paper, we introduce Bayesian Evidential Deep Learning (BEDL), an efficient and generalizable framework for online action detection and uncertainty quantification. Specifically, we combine Bayesian neural networks and evidential deep learning by a teacher-student architecture. The teacher model is built in a Bayesian manner and transfers its mutual information and distribution to the student model through evidential deep learning. In this way, the student model can make accurate online inference while efficiently quantifying the uncertainty. Compared to existing evidential deep learning methods, BEDL estimates uncertainty more accurately by leveraging the Bayesian teacher model. In addition, we designed an attention module for active OAD, which actively selects important features based on the Bayesian mutual information instead of using all the features. We evaluated BEDL on benchmark datasets including THUMPS'14, TVSeries, and HDD. BEDL achieves competitive performance while keeping efficient inference. Extensive ablation studies demonstrate the effectiveness of each component. To verify the uncertainty quantification, we perform experiments of online anomaly detection with different types of uncertainties.

Keywords: Online action detection · Evidential deep learning · Bayesian neural networks · Knowledge distillation · Uncertainty quantification

1 Introduction

Online action detection (OAD) [12] aims at identifying the ongoing action in a streaming video based on the historical observations. Different from the offline setting [54, 69], it only use the past information and makes the prediction as soon as the action takes place. It has many important applications such as autonomous driving [9], visual surveillance [55], and human-robot interaction [25]. OAD is challenging due to the incomplete observations of actions and redundant information among inputs such as background and irrelevant actions. In addition, capturing the predictive uncertainty and generalizing to unseen environments are difficult, which are required by most safety-critical applications such as autonomous driving.

To address these challenges, we introduce Bayesian Evidential Deep Learning (BEDL) for active online action detection and uncertainty quantification. Specifically, we combine Bayesian neural networks and evidential deep learning by a teacher-student architecture. Firstly, combining Bayesian neural networks and evidential deep learning enables accurate and efficient uncertainty quantification. Secondly, we introduce Bayesian mutual information (BMI) for OAD to actively select important features with minimal redundance and irrelevance.

Evidential deep learning (EDL) [1,52,57] is a technique developed for identifying the unknowns by quantifying uncertainty. Specifically, it uses deep neural networks to predict a Dirichlet distribution over the class probabilities, which is treated as an evidence collection process. Then the learned evidence are used to quantify the uncertainty by a single forward pass. EDL is computationally efficient and it has been widely used for open-set recognition [3, 4, 71] as it enables efficient uncertainty quantification without sampling. However, existing EDL methods cannot accurately model the epistemic and aleatoric uncertainty, which are critical as they provide different perspectives and information about the prediction. On the other hand, Bayesian neural networks (BNN) [20,38,47] can accurately estimate both epistemic and aleatoric uncertainty. But BNN confronts intractability of exact posterior inference and expensive sampling process, which prevents it from practical applications.

Therefore, we combine EDL and BNN in this work to take advantages of their respective strengths, i.e. to efficiently and accurately estimate both epistemic and aleatoric uncertainty by a single forward pass. Specifically, we use BNN to augment EDL for uncertainty estimation by a teacher-student model. First, BNN model (teacher model) is built in a Bayesian manner to model the posterior distribution of model parameters. Then we transfer the knowledge of the teacher model to the EDL model (student model) by knowledge distillation (KD). Different from existing KD approaches that distill prediction scores or features, our BEDL distills both the uncertainties and mutual information of teacher model to student model. In this way, the BEDL model inherits the merits of Bayesian neural networks for accurately quantifying both epistemic and aleatoric uncertainty, while preserving its computational efficiency. As a result, the BEDL model can accurately and efficiently estimate both types of uncertainties by a single forward pass. The distilled mutual information as used as for feature selection as below.

To address the background and irrelevant contents issue, we also designed an attention module for the BEDL model based on the distilled Bayesian mutual information (MI). We measure the dependency between the ongoing action and historical features by MI. Then the attention module is learned to generate an attention mask that can select informative features with high MI. In this way, the prediction benefits from better features and the online action detection can be improved.

We evaluated BEDL's accuracy and efficiency on benchmark datasets including THUMOS'14 [31], TVSeries [12], and HDD [50]. We also demonstrated BEDL's other properties such as data-efficiency and generalization by extensive

ablation studies. In addition, we validated the effectiveness of EDL by uncertainty quantification and online anomaly detection.

In summary, the main contributions of this paper are as follows:

- We introduce BEDL based on combining Bayesian neural newtorks and evidential deep learning, which enables efficient and accurate online inference and uncertainty quantification.
- We design a mutual information based attention module for the BEDL model to select informative features that are more discriminant and reliable.
- Our proposed BEDL achieves competitive performance on online action detection benchmark datasets in terms of accuracy and efficiency. And we validated its uncertainty quantification capability.

2 Related Work

Uncertainty quantification. Bayesian neural networks (BNNs) have traditionally been employed for quantifying uncertainty by considering neural network parameters as probabilistic variables and analyzing their posterior distributions. Techniques for uncertainty quantification (UQ) primarily involve Markov Chain Monte Carlo (MCMC) methods [11, 63, 70] and variational inference (VI) methods [19, 42, 43]. Ensemble approaches [26, 27, 40, 58, 59, 64] also serve as effective alternatives for precise UQ. While traditional BNNs offer several benefits, their need for a wide range of parameter samples for Bayesian inference results in inefficiency in UQ. This inefficiency is due to the requirement for several forward passes through the neural network, each utilizing uniquely sampled parameters from the posterior distribution. To address this issue, evidential deep learning approaches assume the parameters of the target distribution follow a conjugate prior distribution, enabling a closed-form expression for uncertainty that facilitates fast inference. Yet, these methods need additional knowledge to accurately learn the parameters of the prior distribution, including out-of-distribution (OOD) information [44, 45], ensemble models [46], and density estimations [7]. Even with these resources, point-estimation-based evidential networks may not effectively measure epistemic uncertainty, as indicated by [56]. Our strategy, therefore, involves utilizing BNNs to guide the training of evidential networks, aiming at enhancing the accuracy of evidential deep learning models while ensuring high efficiency for inference.

Evidential deep learning (EDL). EDL [3, 4, 71] is proposed to address the challenge of making the model know the unknown when the input data is unfamiliar. In addition, it allows uncertainty estimation in a single model and forward pass [57]. Recently, EDL has been widely used for open-set recognition tasks. Bao *et al*. introduced DEAR [3] for open-set action recognition. By formulating action recognition in an EDL framework, the unknown actions can be recognized by computing the predictive uncertainty. Zhao *et al*. proposed multi-label evidential learning (MULE) [71] for open-set action recognition and novelty detection. The proposed evidence debiasing constraint enables the model to handle

general problems of single or multiple actors in the same scene, with simultaneous action(s) by any actor. Later, EDL has been applied for open-set action localization [4,10,23]. By quantifying the frame-level uncertainty, the model can distinguish unknown actions from background video framew. Park *et al*. proposed MEH-HUA [48] for object detection. By aggregating the uncertainty from EDL in a bottom-up order, the context within the image can be better captured for object detection. In this paper, we introduce EDL for online action detection and anomaly detection, which has not been explored yet.

Online action detection. For the model architecture, RNN-based designs [8, 13,15,16,21,24,28,36,41,65,68] are widely adopted because of RNN's temporal modeling capability. Typically, Xu *et al*. [65] proposed temporal recurrent network (TRN) that leverages both the historical information and predicted future features to detect the ongoing action. Thanks to the self-attention mechanism and the parallel computing property, Transformer-based methods [5, 29, 34, 51, 60, 62, 66, 67, 73] become the mainstream for online action detection. Wang [62] proposed OadTR that makes use of both historical information and future prediction. Xu [62] proposed long short-term Transformer (LSTR) that captures both the long-range and short-term dependencies by two memory units. To overcome the latency of feature extraction, [5] proposed E2E-LOAD for end-to-end online action detection. Besides RNN and Transformer, graph modeling is also studied for online action detection [14]. To leverage the video-level annotations instead of the dense frame-level annotations, weakly-supervised methods detection [24, 68] are also explored for OAD.

3 Approach

In this section, we first give an overview of the proposed Bayesian evidential deep learning (§ 3.1) and formulate the online action detection problem (§ 3.2). Then we introduce BEDL's Bayesian teacher model (§ 3.3) and evidential student model (§ 3.4). In the end, we show the training procedures (§ 3.5).

3.1 Overview

The overall framework of BEDL is shown in Figure 1. BEDL is built as a teacher-student framework, which is composed of a Bayesian neural network as the teacher model and an evidential neural network as the student model. Given a streaming video as input, a pretrained backbone first extracts features. Then the Bayesian tearcher model outputs the action predictions and mutual information based on the features, which are used to train the evidential student model by knowledge distillation. During the testing, only the student model is kept to perform fast inference and efficient uncertainty quantification.

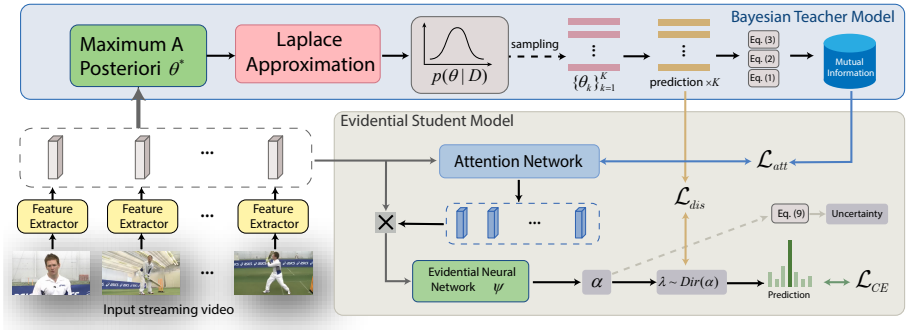


Fig. 1: Overall framework of Bayesian evidential deep learning (BEDL). The input of model is a streaming video. A pretrained backbone is used to extract the features for each frame. During the training, the Bayesian teacher model generates the mutual information (MI) and multiple sets of predictions. The evidential student model leverages MI to train the attention module for feature selection and predictions for uncertainty quantification. During testing, the student model actively selects informative features and quantifies predictive uncertainty accurately and efficiently by a single forward pass.

3.2 Problem formulation

Online action detection(OAD) aims at recognizing the ongoing action in a streaming video with only the past and current observations. Denote the input video as $\boldsymbol{V} = [I_1, I_2, ..., I_T]$, where T is the length of video and I_t denotes the frame at current time t. The online action detection is formulated as a classification problem: $y_t^* = \text{argmax}_c\, p(\hat{y}_t = c|\boldsymbol{V}_t)$, where $\hat{y}_t$ is the prediction, c is the class label, and $\boldsymbol{V}_t = \{I_1, ...I_t\}$ is the available frame set at time t. A feature extractor is used to process each frame and generate the corresponding feature vector. Denote the feature set at time t as $\boldsymbol{F}^t = \{F_1^t, ..., F_t^t\}$. The feature at time i is $F_i^t \in \mathbb{R}^J$, where J is the feature dimension of each frame.

3.3 Bayesian teacher model

To take advantages of Bayesian neural networks for modeling both epistemic and aleatoric uncertainty, we build a Bayesian teacher model (BTM). BTM's objectives include: 1) generating mutual information to train the attention module for feature selection. 2) providing distribution knowledge for the student model for uncertainty quantification.

The mutual information (MI) between past features and ongoing action indicates the relevance of features. Denote a past feature as F_{ij}^t, where $i \in \{1, ..., t\}$ is the time index, $j \in \{1, ..., J\}$ is the feature index within each frame, and t denotes the current time. We aim to obtain the mutual information between F_{ij}^t and the ongoing action y_t. An illustration is shown in Figure 2. To compute mutual information, we take advantage of Bayesian modeling. Different from

point estimation, Bayesian method constructs a posterior distribution of model parameters. By integrating predictions from multiple models, it is less likely to be overfitting and the predictive uncertainty can be accurately quantified. Additionally, Bayesian method is more robust when training data is insufficient. We term the mutual information computed using the Bayesian method Bayesian Mutual Information (BMI).

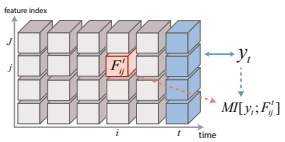


Fig. 2: Illustration of mutual information in OAD.

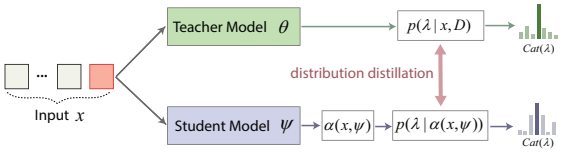


Fig. 3: Illustration of distribution distillation by Bayesian evidential deep learning.

Denote the model parameters of the teacher model as θ and we treat them as probability distributions. Then the BMI between a past feature F_{ij}^t and the ongoing action y_t can be written as:

$$\mathcal{I}[y_t; F_{ij}^t|\mathcal{D}] = \mathcal{H}[y_t|\boldsymbol{F}_{-ij}^t, \mathcal{D}] - \mathcal{H}[y_t|\boldsymbol{F}^t, \mathcal{D}] \tag{1}$$

where $\mathcal{D}$ denotes the training data, $\mathcal{H}$ denotes the entropy, and $\boldsymbol{F}_{-ij}^t$ is the feature set at time t excluding F_{ij}^t, i.e. $\boldsymbol{F}_{-ij}^t = \boldsymbol{F}^t / F_{ij}^t$. By definition, the entropy term is written as:

$$\mathcal{H}[y_t|\boldsymbol{F}^t, \mathcal{D}] = -\sum_{y_t \in \mathcal{Y}} p(y_t|\boldsymbol{F}^t, \mathcal{D}) \log p(y_t|\boldsymbol{F}^t, \mathcal{D}) \tag{2}$$

where $\mathcal{Y} = \{0, 1, ..., C\}$ is the action class set. 0 represents background class and C is the number of action classes. The prediction distribution in Eq. (2) can be computed as:

$$p(y_t|\boldsymbol{F}^t, \mathcal{D}) = \int p(y_t|\boldsymbol{F}^t, \theta) p(\theta|\mathcal{D}) d\theta \approx \frac{1}{K} p(y_t|\boldsymbol{F}^t, \theta_k), \ \ where \ \ \theta_k \sim p(\theta|\mathcal{D}) \tag{3}$$

In Eq. (3), we use the sample average to approximate it since it is impractical to integrate over all the possible parameters. Similarly, we can compute $\mathcal{H}[y_t|\boldsymbol{F}_{-ij}^t, \mathcal{D}]$.

To obtain the posterior distribution $p(\theta|\mathcal{D})$, we perform the Laplace approximation (LA). To reduce the computation cost, we adopt a last-layer Bayesian setting [39]. Only the posterior distribution of last-layer parameters is modeled, while keeping the remaining parameters deterministic. Later, we will show this is efficient and effective. Specifically, we assume the last-layer parameters follows a Gaussian distribution.

Firstly, we train the teacher model under the deterministic setting by a cross entropy loss:

$$\begin{aligned}\theta^* &= \underset{\theta}{\operatorname{argmin}}\, \mathcal{L}(\mathcal{D};\theta) = \underset{\theta}{\operatorname{argmin}} \Big(\sum_{n=1}^{N} l(x_n, y_n; \theta) + r(\theta) \Big) \\ &= \underset{\theta}{\operatorname{argmin}} \Big(\sum_{n=1}^{N} -\log p\big(y_n | f_\theta(x_n)\big) - \log p(\theta) \Big)\end{aligned} \tag{4}$$

where $\mathcal{D} = \{x_n, y_n\}_{n=1}^{N}$ is the training set and $r(\theta)$ is a regularizer such as weigh regularizer (a.k.a. weight decay). So θ^* is indeed a maximum a posteriori (MAP) estimate.

The Laplace approximation uses a second-order Taylor expansion of $\mathcal{L}(\mathcal{D};\theta)$ around θ^* to construct a Gaussian approximation to $p(\theta|\mathcal{D})$:

$$\mathcal{L}(\mathcal{D};\theta) \approx \mathcal{L}(\mathcal{D};\theta^*) + \frac{1}{2}(\theta - \theta^*)^T (\nabla_\theta^2 \mathcal{L}(\mathcal{D};\theta)|_{\theta^*})(\theta - \theta^*) \tag{5}$$

where the first-order term vanishes at θ^*. So the Laplace posterior approximation can be obtained as:

$$p(\theta|\mathcal{D}) \approx \mathcal{N}(\theta^*, \Sigma), \;\; with \;\; \Sigma := (\nabla_\theta^2 \mathcal{L}(\mathcal{D};\theta)|_{\theta^*})^{-1} \tag{6}$$

After the LA, we sample K times from $p(\theta|\mathcal{D})$ to obtain K sets of parameters $\{\theta_1, ..., \theta_K\}$. Then we approximate the prediction distribution in Eq. (3) and further compute the mutual information of each feature. The BMI we computed are based on multiple models from the teacher model. We use BMI to supervise the attention module of student model as the importance of past feature is measured by MI. Different from the traditional MI estimated from probabilistic models such as Uncertainty-OAD [27], BEDL uses BMI and hence is more robust and accurate.

3.4 Evidential student model

Evidential deep learning (EDL). Softmax-based deep neural networks are unable to accurately estimate the predictive uncertainty since the softmax probability is a point estimation of the predictive distribution and it tends to be over-confident in wrong predictions. To address such issue, EDL is proposed by combing Dempster-Shafer Theory [53] and subjective logic [33]. Specifically, it assumes the classification probability follows a prior Dirichlet distribution. The training is considered as a process of collecting evidence. The learned parameters of Dirichlet distribution are used as evidence to compute the predictive uncertainty. In this way, the uncertainty can be estimated in a single forward pass. In this work, we build the evidential student model under the EDL framework so that it can output both epistemic and aleatoric uncertainty in a forward pass.

Distribution distillation. Common knowledge distillation (KD) methods aim at transferring prediction scores, features, or evidence [22], so the student model

cannot estimate the predictive uncertainty. Differently, we transfer the uncertainty of the Bayesian teacher model to the evidential student model, i.e. distilling the ditribution. In this way, the student model can learn from predictions of the teacher model as well as the capability of capturing the uncertainty.

Under the classification setting, the target output of student model y follows a categorical distribution with parameter λ, $y \sim p(y|\lambda) = Cat(\lambda)$. For example, λ represents the probability after the final softmax layer. We treat λ as a random variable and assume it follows Dirichlet distribution, i.e. $\lambda \sim p(\lambda|\alpha(x,\psi)) = Dir(\alpha(x,\theta))$, where α denotes the parameters of the Dirichlet distribution and ψ denotes student model parameters. Similarly, the teacher model has posterior distribution $p(\lambda|x,\mathcal{D}) = \int p(\lambda|x,\theta)p(\theta|\mathcal{D})d\theta$. During the distillation, we transfer teacher posterior distribution $p(\lambda|x,\mathcal{D})$ to the student posterior distribution $p(\lambda|\alpha(x,\psi))$. An illustration is shown in Figure 3. Specifically, we minimize the KL-divergence between these two distributions as below:

$$\begin{aligned}\mathcal{L}_{dis} &= KL(p(\lambda|x,\mathcal{D})||p(\lambda|\alpha(x,\psi))) \\ &\propto -\sum_{c=1}^{C}\log(\Gamma(\alpha_c)) + \log\Gamma(\sum_{c=1}^{C}\alpha_c) - \mathbb{E}_{p(\theta|\mathcal{D})}[\sum_{c=1}^{C}(\alpha_c - 1)\log\lambda_c(x,\theta)]\end{aligned} \tag{7}$$

where C is the number of action classes. Detailed derivation of $\mathcal{L}_{dis}$ can be found in Appendix. Since the different predictions of the teacher model are combined, the model is more robust. The complete distribution distillation algorithm is available in the supplementary.

Mutual information based attention module. The attention module actively selects informative features from the inputs by generating a spatial-temporal attention mask $\boldsymbol{A}_t$ using a fully-connected network and applying to the original features by element-wise product. In this way, irrelevant features are masked out since they have low mutual information to the ongoing action. The training of the attention module is supervised by the BMI $\mathcal{I}_t$ from the teacher model. In this way, the BMI from the Bayesian teacher model is distilled to the student model. By minimizing the mean squared error (MSE) loss $\mathcal{L}_{att}$ between $\boldsymbol{A}_t$ and $\mathcal{I}_t$, the attention module can directly generate the BMI-aware attention mask without computing the BMI. $\mathcal{L}_{att}$ can be written as below:

$$\mathcal{L}_{att} = MSE(\boldsymbol{A}_t, \sigma(\mathcal{I}_t)) = \frac{1}{tJ}\sum_{i,j}||A_{ij}^t - \sigma(\mathcal{I}[y_t; F_{ij}^t|\mathcal{D}])||^2 \tag{8}$$

where $\sigma(\cdot)$ is the sigmoid function.

Uncertainty quantification. After the distillation through evidential deep learning, the Dirichlet distribution of the student model gains the knowledge of the Bayesian teacher model. The uncertainty can be computed efficiently in a closed-form solution by a single forward pass. Specifically, the total uncertainty

and epistemic uncertainty can be quantified as:

$$\begin{aligned} \mathcal{H}[p(y|x,\theta)] &= \sum_{c=1}^{C} \frac{\alpha_c}{\alpha_0} \log \frac{\alpha_c}{\alpha_0}, \text{ where } \alpha_0 = \sum_{c=1}^{C} \alpha_c \\ \mathcal{I}[y;\lambda|\alpha] &= -\sum_{c=1}^{C} \frac{\alpha_c}{\alpha_0} \big(\ln \frac{\alpha_c}{\alpha_0} \Psi(\alpha_c + 1) + \Psi(\alpha_0 + 1) \big) \end{aligned} \tag{9}$$

where $\Psi(\cdot)$ is the dgamma function. To be simple, the aleatoric uncertainty can be computed by taking the difference between total uncertainty and epistemic uncertainty. The uncertainty can be also used to detect the anomaly. Details derivations are available in the supplementary.

3.5 Training

The evidential student model is jointly trained for online action detection, attention distillation, and distribution distillation. Specifically, we adopt a two-strategy. We first train the attention module using the attention loss $\mathcal{L}_{att}$ in Eq. 8. Then we jointly train the evidential student model by total loss $\mathcal{L}$ below:

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda_1 \mathcal{L}_{att} + \lambda_2 \mathcal{L}_{dis} \tag{10}$$

where $\mathcal{L}_{ce}$ is the cross-entropy loss for online action detection. λ_1 and λ_2 are hyper-parameters that emphasize the attention module and distribution distillation. Although the distribution distillation can make the student model perform online action detection, we still perform joint training because the Bayesian teacher model is not perfect and its posterior distribution is approximated, which may introduce error to the student model. The joint training brings two benefits: 1) the online action detection can be improved with the supervision of ground-truth label; 2) the negative log-likelihood $\mathcal{L}_{ce}$ makes the training faster and more stable as indicated in [18]. We demonstrate the two-stage training works better in Sec. 4.6.

4 Experiments

4.1 Datasets and evaluation metrics

THUMOS'14 [31] is a dataset for video-based temporal action localization. We use the validation set with 200 videos for training and test set with 213 videos for evaluation. There are 20 action classes and a background class. We ignore the frames with ambiguous labels. Following the settings in [21, 65], we adopt the mean Average Precision (**mAP**) as the evaluation metric.

TVSeries [12] is a dataset collected from real TV series. It contains 27 episodes with 30 daily actions. It is a challenging dataset due to the viewpoints changing and occlusions in the videos. To counter the imbalanced data distribution, we adopt the mean calibrated Average Precision (**mcAP**) [12] as the

evaluation metric. It is computed as $cAP = \sum_k cPrec(k) \times \mathbb{1}(k)/P$, where $cPrec = TP/(TP + FP/\omega)$, P is the total number of positive frames and $\mathbb{1}(k)$ is an indicator function that is equal to 1 if frame k is a true positive. The mcAP is the mean of calibrated average precision of all action classes.

HRI Driving Dataset (HDD) [50] is a dataset for learning driver behavior in real-life environments. It contains 104 hours of real human driving in the San Francisco Bay Area collected by an instrumented vehicle with different sensors. There are 11 goal-oriented driving actions. Following the settings in [50], we use 100 sessions for training and 37 sessions for testing. And only the sensor data is used as the input. **mAP** is used as the evaluation metric for this dataset.

Table 1: Online anomaly detection results on THUMOS'14 and TVSeries.

Uncertainty	THUMOS'14 (%)	TVSeries (%)
EDL [3]	64.72	42.39
Total	81.04	65.16
Aleatoric	62.51	39.83
Epistemic	**86.48**	**69.44**

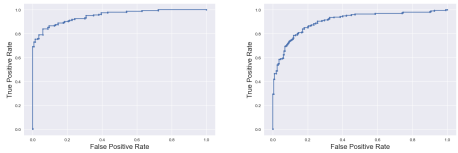


Fig. 4: ROC curves of online anomaly detection using epistemic uncertainty on THUMOS'14 (left) and TVSeries (right).

4.2 Implementation Details

Feature extraction. Following the settings in [15, 65, 66], we use TSN [61] to extract the features for THUMOS'14 and TVSeries. Video frames are extracted at 24 fps and the chunk size is set to 6. To better capture the spatial-temporal dependencies, we adopt the multi-scale vision Transformer [17] to extract RGB features. The optical flow features are extracted with BN-Inception [32]. The backbone is pretrained on ActivityNet [30] and Kinetics-400 [6] separately for evaluation. For HDD dataset, the sensor data is used as the input. More details can be found in the supplementary.

Settings. The BEDL is implemented in PyTorch [49]. The model is trained by the Adam optimizer [37] with a learning rate of 10^{-4} and a weight decay of 5×10^{-5}. The batch size is set to 32. The experiments were conducted on two Nvidia RTX 3090 Ti GPUs. The number of epochs is set to 20. Ablation studies on existing methods are based on their officially released codes. Detailed model architectures are available in the supplementary. And csode will be made publicly available.

4.3 Online action detection results

We evaluate BEDL on three benchmark datasets and make a comparison with other methods in Table 2. From the results, our proposed BEDL achieves state-of-the-art performance on most benchmarks. On THUMOS'14, BEDL achieves 70.6% and 72.7% mAP using ActivityNet and Kinetics pretrained features. On TVSeries, BEDL achieves 88.9% and 90.1% mcAP using ActivityNet and Kinetics features respectively. On HDD, we only use the sensor data as the input and

achieve 33.0% mAP, which is higher that the SOTA MAT [60]. The OAD performance demonstrate the effectiveness of BEDL for accurate online inference. In the following parts, we show other superior properties of BEDL while keeping accurate inference.

Table 2: Experiment results on THUMOS'14, TVSeries and HDD. The results on THUMOS'14 and HDD are reported as mAP (%). The results on TVSeries are reported as mcAP (%). For HDD, ⋆ indicates RGB data is used as the input.

Method	Architecture	THUMOS'14		TVSeries		HDD
		ANet	Kinetics	ANet	Kinetics	Sensor
RED [21]	RNN	45.3	-	79.2	-	27.4
FATS [35]		-	59.0	81.7	84.6	-
TRN [65]		47.2	62.1	83.7	86.2	29.2
IDN [15]		50.0	60.3	84.7	86.1	-
PKD [72]		-	64.5	-	86.4	-
WOAD [24]		-	67.1	-	-	-
OadTR [62]	Transformer	58.3	65.2	85.4	87.2	29.8
*Co*OadTR [29]		56.1	64.2	87.6	87.7	30.6
Colar [67]		59.4	66.9	86.0	88.1	30.6
LSTR [66]		65.3	69.5	88.1	89.1	-
Uncertainty-OAD [27]		66.0	69.9	88.3	89.3	30.1
TeSTra [73]		68.2	71.2	-	-	-
GateHUB [8]		69.1	70.7	88.4	89.6	32.1
MiniROAD [2]		69.3	71.8	88.5	89.6	-
MAT [60]		70.4	71.6	**88.6**	89.7	32.7
E2E-LOAD [5]		-	72.4	-	**90.3**	48.1⋆
BEDL (ours)		**70.6**	**72.7**	**88.9**	90.1	**33.0**

4.4 Uncertainty quantification and online anomaly detection

To verify the uncertainty quantification of BEDL, we quantified different types of uncertainties and perform online anomaly detection on THUMOS'14 and TVSeries. Specially, we divide the data into known classes and unknown classes (anomaly). Unknown classes do not appear in the training set. During testing, both known and unknown data will appear and the model is required to identify if the input belong to known or unknown classes. For THUMOS'14, we treat the "ambiguous" class that are difficult to identify during labeling process as anomaly. For TVSeries, we randomly select 5 out of 30 classes as the anomaly and repeat the process for ten times. Inputs that lead to high predictive uncertainty above the pre-defined threshold are declared as anomalies.

The experiment results are shown in Table 1 and ROC curves are plotted in Figure 4. From the results, BEDL can accurately identify the unknown classes during the testing, which validates the effectiveness of its uncertainty quantification. Since epistemic uncertainty represents the lack of knowledge of the model and it is inversely proportional to the data density, it has better anomaly detec-

tion accuracy compared to aleatoric uncertainty, which represents the noise level of the data. Detailed experiment procedures can be found in the supplementary.

4.5 Computational efficiency and model complexity

We leverage evidential deep learning for BEDL in order to make it more practical for real-time applications. A comparison of computation efficiency and model complexity is shown in Table 3. While keeping a high detection accuracy, our proposed BEDL has much less model complexity and computational cost. And the inference speed of the model is much higher compared to other methods. Recently, [5] proposed an end-to-end OAD framework to avoid the latency from the frame and feature extraction parts, which leads to higher overall inference speed. To make fair comparisons, we used the same feature extraction procedures as prior works and we mainly compare the model inference speed since it only depends on the model architecture and pipeline.

Table 3: Comparison of computation efficiency and model complexity. Our proposed BEDL has less model complexity and computational cost. And the inference speed is much faster than other methods.

Method	Modality	Model		Inference Speed (FPS)			
		Parameter Count	GFLOPs	Optical Flow Computation	RGB Feature Extraction	Flow Feature Extraction	Model
TRN	RGB + Flow	402.9M	1.46	8.1	70.5	14.6	123.3
OadTR		75.8M	2.54	8.1	70.5	14.6	110.0
LSTR		58.0M	7.53	8.1	70.5	14.6	91.6
GateHUB		45.2M	6.98	8.1	70.5	14.6	71.2
MAT		94.6M	-	8.1	70.5	14.6	72.6
BEDL (ours)		19.4M	0.48	8.1	70.5	14.6	**163.2**

4.6 Ablation studies

Training strategies. To demonstrate the effectiveness of two-stage training strategy, we compared it with two other training strategies. One is jointly training the attention module and EDL model by the total loss in Eq. 10. The other is firstly training the attention module by the attention loss and then we freezing the weights of attention module to train the EDL model by the $\mathcal{L}_{att}$ and $\mathcal{L}_{ce}$. We refer the first one as joint training and the second one as separate training. The performance comparison is shown in Table 4. The results show that the two-stage training outperforms the other two strategies on both THUMOS'14 and TVSeries with different features, which demonstrates its effectiveness.

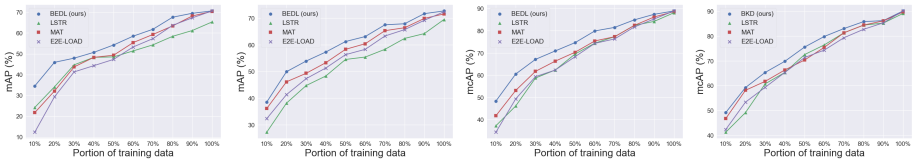
Number of historical frames. At time t, BEDL takes a certain number of past frames to predict the ongoing action. To study the long-range and short-term modeling capability of BEDL, we vary the number of past frames as the input. The experiment results are shown in Table 5. THUMOS'14 and TVSeries

Table 4: Ablation study of training strategies. The two-stage training gives the best performance.

Training	THUMOS'14		TVSeries	
	ActivityNet	Kinetics	AcivityNet	Kinetics
Joint	69.5	70.7	85.4	88.2
Separate	64.9	67.8	81.6	84.0
Two-stage	**70.6**	**72.7**	**88.9**	**90.1**

Table 5: Ablation study of different number of historical frames.

Dataset	Feature	Number of past frames				
		8	16	32	64	128
THUMOS'14	ANet	41.6	59.2	68.7	**70.6**	66.5
	Kinetics	49.3	61.6	71.4	**72.7**	70.0
TVSeries	ANet	63.5	76.7	87.6	**88.9**	84.9
	Kinetics	68.2	79.3	88.5	**90.1**	88.2
HDD	Sensor	30.4	31.8	**33.0**	32.1	30.2



(a) THUMOS'14-ANet **(b)** THUMOS'14-K400 **(c)** TVSeries-ANet **(d)** TVSeries-K400

Fig. 5: Experiment results of training with small-scale data. We reduce the training data from 100% to 10% and compared with LSTR [66], MAT [60], and E2E-LOAD [5]. The results are plotted for THUMOS'14 and TVSeries with both ActivityNet and Kinetics pretrained features. Our proposed BEDL outperforms all others when training data is limited.

are extracted at 6 fps. HDD is extracted at 3 fps. So the optimal number of frames on HDD is 32.

Data-efficiency. By applying the mutual information based attention mechanism, we expect BEDL to be more data-efficient when the amount of training data is limited. We reduce the training data from 100% to 10% and compare with other methods. The results on THUMOS'14 and TVSeries are plotted in Figure 5. When training data is reduced, BEDL has less performance decay and outperforms other methods, which demonstrates that BEDL is more data-efficient.

Loss function. In the second training stage of BEDL, the total loss in Eq. 10 is composed of three terms. λ_1 and λ_2 are the weights for attention loss and distribution distillation loss. To study the balance of all three terms, we train the model with different λs on THUMOS'14 and TVSeries. the comparisons are plotted in Figure 6. We empirically choose $\lambda_1 = 0.4$ and $\lambda_2 = 6$ since they give the best performance.

Generalization. To test the generalization capability of the model, we perform the Cross-View and under occlusion experiments on TVSeries dataset. Following the settings in [27], the training set and test set are from different view angles. For the occlusion, the training data does not contain occlusion and the testing data is occluded. The experiment results and comparison are shown in Table 6. The results show that BEDL generalizes better under different conditions.

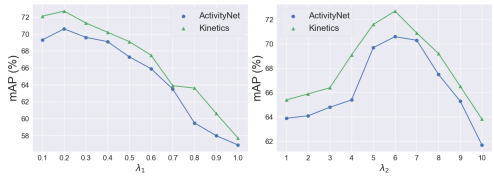


Fig. 6: Ablation study of loss function for λ_1 and λ_2 on THUMOS'14 with ActivityNet and Kinetics features.

Table 6: Cross-view and occlusion experiments results on TVSeries.

Method	CV (%)	Occ. (%)
TRN [65]	65.8	85.2
OadTR [62]	66.2	87.7
U-OAD [27]	67.3	89.5
Colar [67]	66.7	88.3
LSTR [66]	69.5	89.4
TeSTra [73]	70.2	89.9
BEDL (ours)	**74.3**	**91.8**

Attention module. To verify the effectiveness Bayesian mutual information, we first compare with the model without BMI. Specifically, we trained the student model with the same evidential neural network without the attention module. The total loss function is $\mathcal{L} = \mathcal{L}_{ce} + \lambda_2 \mathcal{L}_{dis}$ and we grid-searched λ_2 to obtain the best accuracy. In addiction, we compute the mutual information based on a single prediction instead of BMI and trained the model by the same loss function in Eq. 10. The results are shown in Table 7. From the results, the BEDL with BMI outperforms other two methods, which demonstrates the effectiveness of attention mechanism using BMI.

Last-layer Bayesian. For the teacher model, we adopt the last-layer Bayesian [39] method to reduce the difficulty of training. We also performed the Laplace approximation over all the model parameters. The comparison is shown in Table 8. The full-Bayesian method has consistent improvement on THUMOS-14 and TVSeries. But the LA in training takes much longer time than the last-layer Bayesian and the model needs careful tuning, so we adopt the last-layer Bayesian in the LA process.

Table 7: Effectiveness of BMI. BEDL-No-BMI denotes the model without attention module and BEDL-MI denotes the model with non-Bayesian MI.

Model	THUMOS'14		TVSeries	
	ActivityNet	Kinetics	AcivityNet	Kinetics
BEDL-No-BMI	63.1	64.8	86.5	87.4
BEDL-MI	67.6	68.4	88.0	88.5
BEDL	**70.6**	**72.7**	**88.9**	**90.1**

Table 8: Comparison of full-Bayesian and last-layer Bayesian. Full-Bayesian improves the performance since it models the distribution of all model parameters.

Method	THUMOS-14 (mAP %)		TVSeries (mcAP %)	
	ActivityNet	Kinetics	AcivityNet	Kinetics
Last-layer	70.6	72.7	88.9	90.1
Full-Bayesian	71.1	73.0	90.2	90.3

5 Conclusion

Conclusions. In this paper, we introduce Bayesian evidential deep learning for active online action detection and uncertainty quantification. By combining Bayesian neural networks and evidential deep learning, BEDL can accurately and efficiently quantify predictive uncertainty and make online inference.

References

1. Amini, A., Schwarting, W., Soleimany, A., Rus, D.: Deep evidential regression. Advances in Neural Information Processing Systems **33**, 14927–14937 (2020) 2
2. An, J., Kang, H., Han, S.H., Yang, M.H., Kim, S.J.: Miniroad: Minimal rnn framework for online action detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10341–10350 (2023) 11
3. Bao, W., Yu, Q., Kong, Y.: Evidential deep learning for open set action recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13349–13358 (2021) 2, 3, 10
4. Bao, W., Yu, Q., Kong, Y.: Opental: Towards open set temporal action localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2979–2989 (2022) 2, 3, 4
5. Cao, S., Luo, W., Wang, B., Zhang, W., Ma, L.: E2e-load: End-to-end long-form online action detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 10422–10432 (October 2023) 4, 11, 12, 13
6. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017) 10
7. Charpentier, B., Zügner, D., Günnemann, S.: Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts. Advances in Neural Information Processing Systems **33**, 1356–1367 (2020) 3
8. Chen, J., Mittal, G., Yu, Y., Kong, Y., Chen, M.: Gatehub: Gated history unit with background suppression for online action detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19925–19934 (2022) 4, 11
9. Chen, L., Ma, N., Wang, P., Li, J., Wang, P., Pang, G., Shi, X.: Survey of pedestrian action recognition techniques for autonomous driving. Tsinghua Science and Technology **25**(4), 458–470 (2020) 1
10. Chen, M., Gao, J., Yang, S., Xu, C.: Dual-evidential learning for weakly-supervised temporal action localization. In: European Conference on Computer Vision. pp. 192–208. Springer (2022) 4
11. Chen, T., Fox, E., Guestrin, C.: Stochastic gradient hamiltonian monte carlo. In: International conference on machine learning. pp. 1683–1691. PMLR (2014) 3
12. De Geest, R., Gavves, E., Ghodrati, A., Li, Z., Snoek, C., Tuytelaars, T.: Online action detection. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14. pp. 269–284. Springer (2016) 1, 2, 9
13. De Geest, R., Tuytelaars, T.: Modeling temporal structure with lstm for online action detection. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1549–1557. IEEE (2018) 4
14. Elahi, G.M.E., Yang, Y.H.: Online temporal classification of human action using action inference graph. Pattern Recognition **132**, 108972 (2022) 4
15. Eun, H., Moon, J., Park, J., Jung, C., Kim, C.: Learning to discriminate information for online action detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 809–818 (2020) 4, 10, 11
16. Eun, H., Moon, J., Park, J., Jung, C., Kim, C.: Temporal filtering networks for online action detection. Pattern Recognition **111**, 107695 (2021) 4
17. Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., Feichtenhofer, C.: Multiscale vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6824–6835 (2021) 10

18. Fathullah, Y., Xia, G., Gales, M.J.: Logit-based ensemble distribution distillation for robust autoregressive sequence uncertainties. In: Uncertainty in Artificial Intelligence. pp. 582–591. PMLR (2023) 9
19. Franchi, G., Bursuc, A., Aldea, E., Dubuisson, S., Bloch, I.: Encoding the latent posterior of bayesian neural networks for uncertainty quantification. arXiv preprint arXiv:2012.02818 (2020) 3
20. Gal, Y., Ghahramani, Z.: Bayesian convolutional neural networks with bernoulli approximate variational inference. arXiv preprint arXiv:1506.02158 (2015) 2
21. Gao, J., Yang, Z., Nevatia, R.: Red: Reinforced encoder-decoder networks for action anticipation. arXiv preprint arXiv:1707.04818 (2017) 4, 9, 11
22. Gao, J., Chen, M., Xu, C.: Collecting cross-modal presence-absence evidence for weakly-supervised audio-visual event perception. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18827–18836 (2023) 7
23. Gao, J., Chen, M., Xu, C.: Vectorized evidential learning for weakly-supervised temporal action localization. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023) 4
24. Gao, M., Zhou, Y., Xu, R., Socher, R., Xiong, C.: Woad: Weakly supervised online action detection in untrimmed videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1915–1923 (2021) 4, 11
25. Goodrich, M.A., Schultz, A.C., et al.: Human–robot interaction: a survey. Foundations and Trends® in Human–Computer Interaction **1**(3), 203–275 (2008) 1
26. Guo, H., Aved, A., Roller, C., Ardiles-Cruz, E., Ji, Q.: Video-based complex human event recognition with a probabilistic transformer. In: Geospatial Informatics XIII. vol. 12525, pp. 184–192. SPIE (2023) 3
27. Guo, H., Ren, Z., Wu, Y., Hua, G., Ji, Q.: Uncertainty-based spatial-temporal attention for online action detection. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV. pp. 69–86. Springer (2022) 3, 7, 11, 13, 14
28. Han, Y., Tan, S.: Twinlstm: Two-channel lstm network for online action detection. In: 2022 26th International Conference on Pattern Recognition (ICPR). pp. 3310–3317. IEEE (2022) 4
29. Hedegaard, L., Bakhtiarnia, A., Iosifidis, A.: Continual transformers: Redundancy-free attention for online inference. arXiv preprint arXiv:2201.06268 (2022) 4, 11
30. Heilbron, F.C., Escorcia, V., Ghanem, B., Niebles, J.C.: Activitynet: A large-scale video benchmark for human activity understanding. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR). pp. 961–970. IEEE (2015) 10
31. Idrees, H., Zamir, A.R., Jiang, Y.G., Gorban, A., Laptev, I., Sukthankar, R., Shah, M.: The thumos challenge on action recognition for videos "in the wild". Computer Vision and Image Understanding **155**, 1–23 (2017) 2, 9
32. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. pmlr (2015) 10
33. Jøsang, A.: Subjective logic, vol. 3. Springer (2016) 7
34. Kim, Y.H., Kang, H., Kim, S.J.: A sliding window scheme for online temporal action localization. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIV. pp. 653–669. Springer (2022) 4
35. Kim, Y.H., Nam, S., Kim, S.J.: Temporally smooth online action detection using cycle-consistent future anticipation. Pattern Recognition **116**, 107954 (2021) 11

36. Kim, Y.H., Nam, S., Kim, S.J.: 2pesnet: Towards online processing of temporal action localization. Pattern Recognition **131**, 108871 (2022) 4
37. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 10
38. Kingma, D.P., Salimans, T., Welling, M.: Variational dropout and the local reparameterization trick. Advances in neural information processing systems **28** (2015) 2
39. Kristiadi, A., Hein, M., Hennig, P.: Being bayesian, even just a bit, fixes overconfidence in relu networks. In: International conference on machine learning. pp. 5436–5446. PMLR (2020) 6, 14
40. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles (2016), http://arxiv.org/abs/1612.01474 3
41. Li, Y., Lan, C., Xing, J., Zeng, W., Yuan, C., Liu, J.: Online human action detection using joint classification-regression recurrent neural networks. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14. pp. 203–220. Springer (2016) 4
42. Louizos, C., Welling, M.: Multiplicative normalizing flows for variational bayesian neural networks. In: International Conference on Machine Learning. pp. 2218–2227. PMLR (2017) 3
43. Maddox, W.J., Izmailov, P., Garipov, T., Vetrov, D.P., Wilson, A.G.: A simple baseline for bayesian uncertainty in deep learning. In: Wallach, H., Larochelle, H., Beygelzimer, A., e-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 13132–13143. Curran Associates, Inc. (2019), http://papers.nips.cc/paper/9472-a-simple-baseline-for-bayesian-uncertainty-in-deep-learning.pdf 3
44. Malinin, A., Gales, M.: Predictive uncertainty estimation via prior networks. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 31, pp. 7047–7058. Curran Associates, Inc. (2018), http://papers.nips.cc/paper/7936-predictive-uncertainty-estimation-via-prior-networks.pdf 3
45. Malinin, A., Gales, M.: Reverse kl-divergence training of prior networks: Improved uncertainty and adversarial robustness. Advances in Neural Information Processing Systems **32** (2019) 3
46. Malinin, A., Mlodozeniec, B., Gales, M.: Ensemble distribution distillation. arXiv preprint arXiv:1905.00076 (2019) 3
47. Molchanov, D., Ashukha, A., Vetrov, D.: Variational dropout sparsifies deep neural networks. In: International Conference on Machine Learning. pp. 2498–2507. PMLR (2017) 2
48. Park, Y., Choi, W., Kim, S., Han, D.J., Moon, J.: Active learning for object detection with evidential deep learning and hierarchical uncertainty aggregation. In: The Eleventh International Conference on Learning Representations (2022) 4
49. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017) 10
50. Ramanishka, V., Chen, Y.T., Misu, T., Saenko, K.: Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7699–7707 (2018) 2, 10

51. Rangrej, S.B., Liang, K.J., Hassner, T., Clark, J.J.: Glitr: Glimpse transformers with spatiotemporal consistency for online action prediction. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3413–3423 (2023) 4
52. Sensoy, M., Kaplan, L., Kandemir, M.: Evidential deep learning to quantify classification uncertainty. Advances in neural information processing systems **31** (2018) 2
53. Sentz, K., Ferson, S.: Combination of evidence in dempster-shafer theory (2002) 7
54. Shou, Z., Wang, D., Chang, S.F.: Temporal action localization in untrimmed videos via multi-stage cnns. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1049–1058 (2016) 1
55. Sultani, W., Chen, C., Shah, M.: Real-world anomaly detection in surveillance videos. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6479–6488 (2018) 1
56. Ulmer, D.: A survey on evidential deep learning for single-pass uncertainty estimation. arXiv preprint arXiv:2110.03051 (2021) 3
57. Ulmer, D., Hardmeier, C., Frellsen, J.: Prior and posterior networks: A survey on evidential deep learning methods for uncertainty estimation. Transactions on Machine Learning Research (2023) 2, 3
58. Valdenegro-Toro, M.: Deep sub-ensembles for fast uncertainty estimation in image classification. arXiv preprint arXiv:1910.08168 (2019) 3
59. Wang, H., Ji, Q.: Diversity-enhanced probabilistic ensemble for uncertainty estimation. In: Uncertainty in Artificial Intelligence. pp. 2214–2225. PMLR (2023) 3
60. Wang, J., Chen, G., Huang, Y., Wang, L., Lu, T.: Memory-and-anticipation transformer for online action understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 13824–13835 (October 2023) 4, 11, 13
61. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: European conference on computer vision. pp. 20–36. Springer (2016) 10
62. Wang, X., Zhang, S., Qing, Z., Shao, Y., Zuo, Z., Gao, C., Sang, N.: Oadtr: Online action detection with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7565–7575 (2021) 4, 11, 14
63. Welling, M., Teh, Y.W.: Bayesian learning via stochastic gradient langevin dynamics pp. 681–688 (2011) 3
64. Wen, Y., Tran, D., Ba, J.: Batchensemble: an alternative approach to efficient ensemble and lifelong learning. arXiv preprint arXiv:2002.06715 (2020) 3
65. Xu, M., Gao, M., Chen, Y.T., Davis, L.S., Crandall, D.J.: Temporal recurrent networks for online action detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5532–5541 (2019) 4, 9, 10, 11, 14
66. Xu, M., Xiong, Y., Chen, H., Li, X., Xia, W., Tu, Z., Soatto, S.: Long short-term transformer for online action detection. Advances in Neural Information Processing Systems **34**, 1086–1099 (2021) 4, 10, 11, 13, 14
67. Yang, L., Han, J., Zhang, D.: Colar: Effective and efficient online action detection by consulting exemplars. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3160–3169 (2022) 4, 11, 14
68. Ye, N., Zhang, X., Yan, D., Dong, W., Yan, Q.: Scoad: Single-frame click supervision for online action detection. In: Proceedings of the Asian Conference on Computer Vision. pp. 2156–2171 (2022) 4

69. Zeng, R., Huang, W., Tan, M., Rong, Y., Zhao, P., Huang, J., Gan, C.: Graph convolutional networks for temporal action localization. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 7094–7103 (2019) 1
70. Zhang, R., Li, C., Zhang, J., Chen, C., Wilson, A.G.: Cyclical stochastic gradient mcmc for bayesian deep learning. arXiv preprint arXiv:1902.03932 (2019) 3
71. Zhao, C., Du, D., Hoogs, A., Funk, C.: Open set action recognition via multi-label evidential learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22982–22991 (2023) 2, 3
72. Zhao, P., Xie, L., Zhang, Y., Wang, Y., Tian, Q.: Privileged knowledge distillation for online action detection. arXiv preprint arXiv:2011.09158 (2020) 11
73. Zhao, Y., Krähenbühl, P.: Real-time online video detection with temporal smoothing transformers. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIV. pp. 485–502. Springer (2022) 4, 11, 14