Uncertainty-Guided Probabilistic Transformer for Complex Action Recognition

Hongji Guo, Hanjing Wang, Qiang Ji
Rensselaer Polytechnic Institute, Troy, NY 12180
{guoh11, wangh36, jiq}@rpi.edu

Abstract

A complex action consists of a sequence of atomic actions that interact with each other over a relatively long period of time. This paper introduces a probabilistic model named Uncertainty-Guided Probabilistic Transformer (UGPT) for complex action recognition. The self-attention mechanism of a Transformer is used to capture the complex and long-term dynamics of the complex actions. By explicitly modeling the distribution of the attention scores, we extend the deterministic Transformer to a probabilistic Transformer in order to quantify the uncertainty of the prediction. The model prediction uncertainty is used to improve both training and inference. Specifically, we propose a novel training strategy by introducing a majority model and a minority model based on the epistemic uncertainty. During the inference, the prediction is jointly made by both models through a dynamic fusion strategy. Our method is validated on the benchmark datasets, including Breakfast Actions, MultiTHUMOS, and Charades. The experiment results show that our model achieves the state-of-the-art performance under both sufficient and insufficient data.

1. Introduction

In general, a complex action refers to a high-level activity like "making a sandwich", which consists of a sequence of atomic actions such as "cutting bun" and "smearing butter" as illustrated in Fig. 1. In this paper, we deal with complex action recognition. It has many applications such as visual surveillance [23], human-robot interactions [24], and sports analysis [33]. Complex action recognition is challenging because of the following reasons: (1) complex actions have relatively long temporal durations, which makes it difficult for conventional dynamic models to capture the long-range dependencies; (2) people perform the same complex actions differently, which causes a large intra-class variation; and (3) background and irrelevant frames contained in the video may cause difficulties for the recognition.

Most existing approaches directly recognize the complex actions without explicitly considering the underlying dynamics [1, 6, 29, 42]. In fact, the information of atomic actions and their interactions are often ignored. However, the underlying interactions among atomic actions over time are crucial for understanding complex actions [21, 44, 47]. Furthermore, complex actions may last for a long period, traditional sequence modeling architectures such as recurrent neural network [30], long short-term memory [12] and gated recurrent unit network [5] may not effectively and efficiently handle the long-range dependencies.

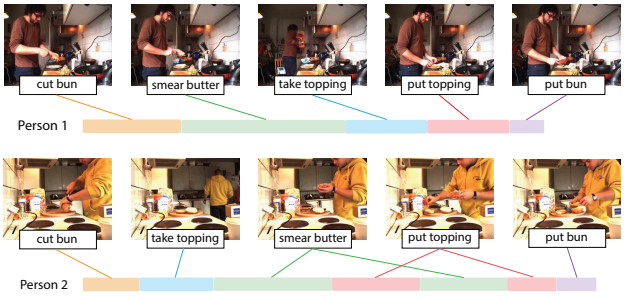

Figure 1. Two samples of "making a sandwich" in Breakfast Actions [18]. A complex action consists of a sequence of atomic actions, which may vary in durations, orders, etc.

In this paper, we propose to use a Transformer [36] as the backbone of our model to explicitly capture the long-term dependencies among atomic actions. Transformers were first proposed for the language translation task to capture the dependencies among words. The self-attention mechanism, which is the core of a Transformer, is now widely used for computer vision tasks such as image generation [45], object detection [20], and group activity recognition [8]. Considering its capabilities of capturing complex and long-range dynamics, Transformer is suitable for modeling complex actions.

While powerful, the conventional Transformer cannot effectively quantify its prediction uncertainty, which is essential to improving the model performance under noisy and imbalanced data distribution. To address this issue, we introduce a probabilistic Transformer. Specifically, we treat the attention scores of a Transformer as random variables to capture the stochastic dependencies and uncertainty in the inputs. We further propose to employ the negative log-

likelihood loss function to train a multilayer perceptron to produce the distribution parameters for the attention scores. The probabilistic attention scores allow us to accurately quantify the epistemic uncertainties of the model prediction. Guided by the prediction uncertainty, we introduce a novel training and inference strategy, whereby we train two models that respectively focus on low-uncertainty samples and high-uncertainty samples, which we refer to as majority model and minority model. During the inference, the two models are combined dynamically based on the uncertainty of the input to perform the final prediction. Experiments show the proposed probabilistic Transformer achieves state of the art performance and is robust under noisy and insufficient data.

The main contributions of this paper are summarized as:

- We propose to exploit the self-attention mechanism of a Transformer to capture the long-term and complex dynamics for complex actions.
- To model the stochasticity in the data and in the complex action, we introduce the probabilistic Transformer that allows accurately quantifying the epistemic uncertainty of the prediction.
- Based on the estimated epistemic uncertainty, we propose a novel strategy for both model training and inference by introducing a majority model and minority model, which improves both the model prediction accuracy and robustness.
- Our method achieved SOTA performance on benchmark datasets, including Breakfast Actions, MultiTHUMOS, and Charades under both sufficient or insufficient training data.

2. Related Work

Complex Action Recognition. Complex action recognition, also known as long-term action recognition, aims at recognizing complex actions in minutes-long videos. It has been an important research topic for many years. Kuehne *et al*. [18] used an HMM to recover the syntax and semantics of complex actions. Tang *et al*. [34] utilized variable-duration HMM to handle the highly varying videos for complex event detection. Wang *et al*. [37] proposed a latent hierarchical (LHM) to model the decomposition of complex actions in a hierarachical way. Zhang *et al*. [21] introduced interval temporal Bayesian network (ITBN) to capture temporal dependencies among time intervals. Liu *et al*. [19] proposed a latent task learning framework with privileged information (LTL-PI) by using the probability of atomic actions. Hussein *et al*. [13] proposed VideoGraph to learn the underlying temporal structure of complex actions by representing human actions as undirected graphs. Hussein *et al*. [14] proposed a Timeception layer with multi-scale temporal convolution kernels to deal with the variational length of atomic actions. Zhou *et al*. [50] proposed a graph-based high-order relation modeling (GHRM) method for long-term action recognition. By combining a Temporal-GHRM and a Semantic-GHRM, the model can well captured the local relations among atomic actions as well as the global dependencies. Also, unsupervised method [26] and transfer learning [19] are also explored in this area.

Probabilistic Dynamics Modeling. Instead of deterministic settings, probabilistic methods model the distribution of the model [35]. It has been applied for various dynamic models. Chien *et al*. [4] proposed a Bayesian RNN for speech recognition. Zhao *et al*. [48] proposed a Bayesian graph convolution LSTM for skeleton-based action recognition to capture the stochasticity and variation in the data. Xue *et al*. [41] proposed a Bayesian Transformer under a full Bayesian learning framework, which aims to mitigate the over-fitting problem and improve the generalization performance. Zhang *et al*. [46] proposed Bayesian attention belief networks based on the self-attention mechanism. The unnormalized attention scores are modeled by a Gamma distributions, which are approximated by the Weibull distributions. Our work is different from [41] and [46] as: (1) we only model the unnormalized scaled dot-product attentions as Gaussian distributions instead of a full Bayesian setting and we assume these distributions are independent; (2) we quantify the epistemic uncertainty and use it to guide the training and inference.

Uncertainty quantification and applications. Under the Bayesian setting, both the epistemic uncertainty and aleatoric uncertainty can be quantified by various methods such as sampling-based method [31], dropout method [7]. Nowadays, the quantified uncertainty shows great potential for computer vision tasks [16]. Subedar *et al*. [32] used uncertainty to help the fusion of visual modality and audio modality for audiovisual activity recognition. Chang *et al*. [2] modeled the data uncertainty to reduce the adverse effects of noisy samples for face recognition. Wang *et al*. [40] utilized the data uncertainty to guide the data selection of multi-phase training for semi-supervised object detection. Yang *et al*. [43] proposed Uncertainty-Guided Transformer Reasoning for camouflaged detection. An uncertainty mask that assigns higher probability to uncertain regions is generated to guide the training for reasoning the target regions. For our work, we use the epistemic uncertainty to guide both the training and inference of complex action recognition.

3. Method

In this section, we first briefly introduce the deterministic Transformer. We then introduce our proposed Uncertainty-Guided Probabilistic Transformer (UGPT) from the follow-

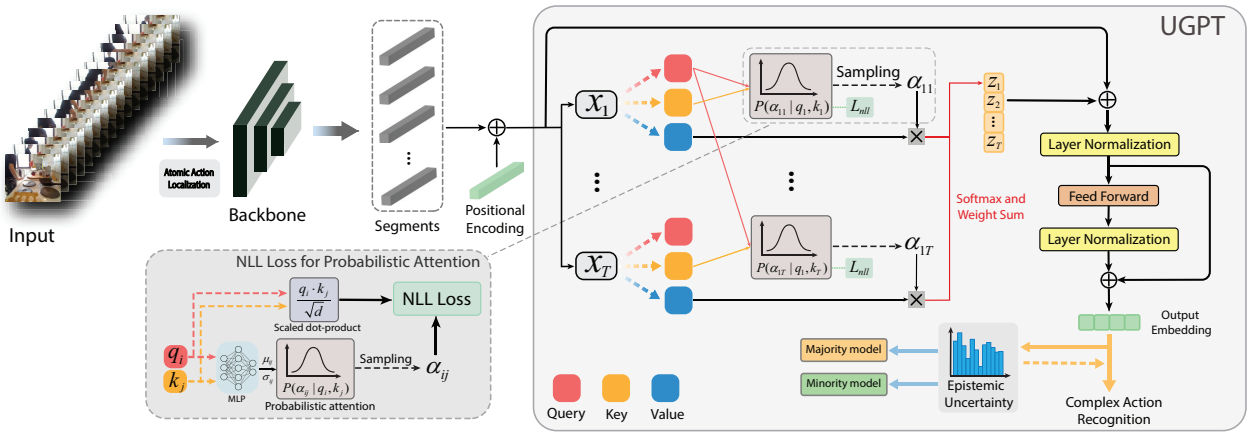

Figure 2. **Framework of Uncertainty-Guided Probabilistic Transformer (UGPT).** The input of our model is a video (sequence). Firstly, an atomic action localization module gives a coarse temporal segmentation of atomic actions. Then a CNN-based backbone is used to extract features for each segment. After adding the positional encoding, the extracted features are fed into the UGPT. Different from deterministic setting, the attentions of our probabilistic Transformer are sampled from Gaussian distributions with a NLL loss. The output embeddigns of the Transformer are used to perform the classification and estimate the epistemic uncertainty, which is further utizlied to guide both the training and the inference.

ing aspects: (1) overall framework; (2) probabilistic attention; (3) uncertainty quantification; (4) probabilistic Transformer training; (5) UGPT inference.

3.1. Deterministic Transformer

Main components. Transformer [36] is a sequence to sequence model initially proposed for language translation. It can effectively capture the long-term dependencies among the inputs thanks to its self-attention mechanism. Here we briefly introduce the main components of the conventional Transformer, which is in deterministic setting.

Given a sequence of tokens as input, positional encoding is firstly added to each input in order to retain the order information of these tokens. Following [36], the positional encoding is computed as:

$$\begin{aligned} PE_{(pos,2i)} &= sin(pos/10000^{2i/C}) \\ PE_{(pos,2i+1)} &= cos(pos/10000^{2i/C}) \end{aligned} \tag{1}$$

where pos is the input position and i is the dimension index.

Each token is then linearly projected to a query (q), a key (k) and a value (v). The dependency between the ith and jth input is measured by the scaled dot-product attention:

$$\alpha_{ij} = \frac{q_i \cdot k_j}{\sqrt{d}} \tag{2}$$

where d is the dimension of queries and keys.

After that, the embedding of the ith input z_i is computed as the attention-weighted sum of values:

$$z_i = \sum_{j=1}^{T} \frac{\alpha_{ij}}{\sum_{j'=1}^{T} \alpha_{ij'}} v_j \tag{3}$$

where T is the length of the sequence.

In this way, the dependency between every pair of inputs is captured even they are far away. Also, this mechanism enables the parallel computing as all the input tokens can be processed simultaneously instead of using a recurrent architecture like RNN or LSTM. Under the deterministic setting, all the parameters are learned during the training and fixed during the inference.

Motivation. Complex actions are composed of atomic actions sequentially aligned in the temporal, much like visual sentences. While a sentence is determined by a sequence of words and their interactions, a complex action analogically is composed of a sequence of atomic actions and their interactions over time. Systematically modelling and capturing the long-term dependencies among atomic actions is essential to accurate recognition of complex actions. A Transformer is hence the natural choice for effectively capturing such dependencies.

Furthermore, considering the uncertainties in the data as well as in the complex actions, the model should be able to effectively capture these uncertainties and propagate them to accurately quantify the confidence of its prediction.

3.2. Uncertainty-Guided Probabilistic Transformer

3.2.1 Overall framework

The overall framework of the proposed Uncertainty-Guided Probabilistic Transformer is shown in Fig. 2. The input of our model is a video (sequence). Firstly, an atomic action localization module [3] is applied to provide a coarse temporal localization of the atomic actions. Then, we extract features for each segment by a CNN-based backbone [11].

These features are used as a token for an atomic action and is fed to the Transformer. To keep the sequential order information, the positional encoding computed by Eq. (1) is added to input tokens. After the positional encoding, the new tokens are fed to our probabilistic Transformer and output high-level embeddings. Then, the output embeddings go through a linear classifier to output probability vectors for classification. The probability vectors are fed into an uncertainty quantification module to generate the epistemic uncertainty (defined in Sec. 3.2.3) for each input. We train two models separately using two different uncertainty weighted loss functions. One model assigns larger weight to low-uncertainty data to emphasize majority of the data, which refers as "majority model". The other assigns larger weights to high-uncertainty data to emphasize minority of the data, which refers as "minority model". In the end, we combine the two models dynamically to make the final prediction.

3.2.2 Probabilistic attention

In the conventional Transformer, the attention of query q_i and key k_j is computed deterministically by Eq. (2), which we refer as deterministic attention. For complex action recognition, the input tokens represent atomic actions. These atomic actions have large variations even within the same class as people perform these actions in different ways, which leads to varied temporal durations, orders, etc. The deterministic attention cannot capture the noise and distribution of the input, and the conventional Transformer is unable to effectively quantify it's prediction uncertainty. To address this issue, we introduce the probabilistic attention. Specifically, we assume α_{ij} follows a Gaussian distribution: $\alpha_{ij} \sim \mathcal{N}(\mu_{ij}, \sigma_{ij}^2)$. The mean and variance of this Gaussian distribution is computed using q_i and k_j through a multilayer perceptron: $\mu_{ij}, \sigma_{ij}^2 = MLP(q_i, k_j)$. Thus, μ_{ij} and σ_{ij} are probabilistic parameters, which are stochastic during both training and inference. Besides these probabilistic parameters, we treat all other model parameters deterministic as usual and denote them as Θ. To train these probabilistic parameters through the gradient descent [25], we adopt the reparameterization trick [17] to perform the forward process of the probabilistic attention as follow:

$$\alpha_{ij} = \mu_{ij} + \sigma_{ij}\epsilon, \quad \epsilon \sim \mathcal{N}(0, 1) \tag{4}$$

where ϵ is a random number sampled from a standard Gaussian distribution. After training, we obtain the trained deterministic parameters, which we denote as Θ^*.

The probabilistic attention allows effectively capturing the input noise and distribution, through which we can compute the probabilistic distribution of target y' given an input

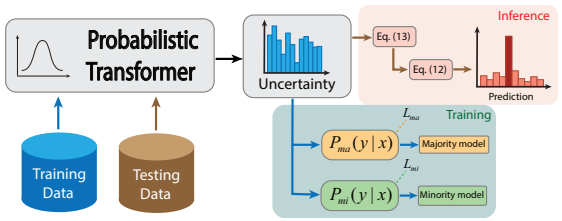

Figure 3. **Uncertainty guiding strategy.** During training, the majority model and minority mode are trained separately based on the quantified uncertainty. During inference, two models are combined dynamically to make the prediction.

$\boldsymbol{X'}$ as follows:

$$\begin{aligned} P(y'|\boldsymbol{X'}, \Theta^*) &= E_{\boldsymbol{\alpha} \sim p(\boldsymbol{\alpha}|\boldsymbol{X'}, \Theta^*)}[P(y'|\boldsymbol{X'}, \Theta^*, \boldsymbol{\alpha})] \\ &= \int_{\boldsymbol{\alpha}} P(y'|\boldsymbol{X'}, \Theta^*, \boldsymbol{\alpha}) p(\boldsymbol{\alpha}|\boldsymbol{X'}, \Theta^*) d\boldsymbol{\alpha} \end{aligned} \tag{5}$$

where $\boldsymbol{\alpha} = \{\alpha_{ij} | i, j \in \{1, ..., T\}\}$ represents all the probabilistic attentions of input $\boldsymbol{X'}$.

As the attention between a query and a key capture the pairwise dependency, we assume all the probabilistic attentions are independent of each other. In this way, we have $p(\boldsymbol{\alpha}|\boldsymbol{X'}, \Theta^*) = \prod_{i,j} p(\alpha_{ij}|\boldsymbol{X'}, \Theta^*)$. However, directly computing the integration of $\boldsymbol{\alpha}$ is intractable. In this work, we sample $\boldsymbol{\alpha}$ from $p(\boldsymbol{\alpha}|\boldsymbol{X'}, \Theta^*)$ for K times to approximate this probabilistic distribution. Specifically, we sample all α_{ij} individually as we assume they are independent. Thus, the complex action recognition is formulated as:

$$y^* = \underset{y'}{\operatorname{argmax}} \frac{1}{K} \sum_{k=1}^{K} P(y'|\boldsymbol{X'}, \Theta^*, \boldsymbol{\alpha_k}) \tag{6}$$

where K is the number of samples to generate given one input, $\boldsymbol{\alpha_k}$ represents the probabilistic attentions of kth sample and $(\boldsymbol{X'}, y')$ is a test input.

3.2.3 Uncertainty quantification

In this part, we introduce the uncertainty in complex action recognition and its quantification method. Complex action recognition is a classification task, the output is the conditional probability distribution $P(y'|\boldsymbol{X'}, \Theta^*)$ over a set of action classes. From $P(y'|\boldsymbol{X'}, \Theta^*)$, we can effectively quantify the prediction uncertainty. There are two main types of uncertainty: epistemic uncertainty and aleatoric uncertainty, which capture the model uncertainty and data uncertainty respectively. These two uncertainties compose the total uncertainty. To quantify the total uncertainty, we compute the entropy of the prediction distribution:

$$\mathcal{H}[P(y'|\boldsymbol{X'}, \Theta^*)] = -\sum_{y' \in \mathcal{Y}} P(y'|\boldsymbol{X'}, \Theta^*) log P(y'|\boldsymbol{X'}, \Theta^*) \tag{7}$$

where $\mathcal{Y}$ represents the action class set.

Also, the total uncertainty is the sum of epistemic uncertainty and aleatoric uncertainty:

$$\begin{aligned}\mathcal{H}\left[P(y'|\boldsymbol{X'},\Theta^*)\right] &= \mathcal{I}\left[y',\boldsymbol{\alpha}|\boldsymbol{X'},\Theta^*\right] \\ &+ \mathbb{E}_{P\left(\boldsymbol{\alpha}|\boldsymbol{X'},\Theta^*\right)}\left[\mathcal{H}[P(y'|\boldsymbol{X'},\boldsymbol{\alpha})]\right]\end{aligned} \tag{8}$$

To refine our model for a better learning and inference, we utilize the epistemic uncertainty in this work, which is the first term in Eq. (8). It is related to model prediction and measures the model uncertainty. It is caused by the limitations of the model. High epistemic uncertainty implies that the model parameters are inadequately estimated because of the lack of data. For complex action recognition, the epistemic uncertainty measures the model uncertainty of complex action predictions. It can give us a measure whether the model is making good predictions, which is useful for the refinement of the model. We utilize this property of epistemic uncertainty to guide the probabilistic Transformer training and inference. During the inference, for each input sample, we generate K $\boldsymbol{\alpha}$s, which further generate multiple outputs. Then the epistemic uncertainty can be quantified as:

$$\begin{aligned}\mathcal{I}(\boldsymbol{\alpha},y'|\boldsymbol{X'},\Theta^*) &= \mathcal{H}[\frac{1}{K}\sum_{k=1}^{K}P(y'|\boldsymbol{X'},\Theta^*,\boldsymbol{\alpha_k})] \\ &-\frac{1}{K}\sum_{k=1}^{K}\mathcal{H}[P(y'|\boldsymbol{X'},\Theta^*,\boldsymbol{\alpha_k})]\end{aligned} \tag{9}$$

where K is the number of samples generated from one input and $\mathcal{H}$ is the entropy.

3.2.4 Probabilistic Transformer Training

Before we discuss the procedure for training the probabilistic Transformer, we first introduce the training of the neural network that outputs the probabilistic attention distribution. To produce the distribution of the probabilistic attention, we propose to use the negative log-likelihood (NLL) loss to train the neural network as illustrated at the bottom left of Fig. 2. The idea is that we use the neural network to learn the parameters of the distribution and constrain the probabilistic attention with the scaled dot-product attention at the same time. Specifically, we use a three-layer MLP to output the mean and variance of the distribution of probabilistic attention. And the negative log-likelihood loss is defined as:

$$\begin{aligned}\mathcal{L}_{nll} = \sum_{n=1}^{N}\sum_{i,j} log\frac{1}{\sqrt{2\pi\sigma_{ij}^2(\boldsymbol{X_n})}} \\ \times exp(-\frac{(\alpha_{ij}(\boldsymbol{X_n})-q_i(\boldsymbol{X_n})\cdot k_j(\boldsymbol{X_n})/\sqrt{d})^2}{2\sigma_{ij}^2(\boldsymbol{X_n})})\end{aligned} \tag{10}$$

Algorithm 1 UGPT Training

Input: $\mathcal{D}=\{\boldsymbol{X_n}\in\mathbb{R}^{T\times d},y_n\}_{n=1}^{N}$: training data
Output: Θ: model parameters

for $\{\boldsymbol{X_n},y_n\}$ in $\mathcal{D}$ **do**
  Compute $\boldsymbol{q_n},\boldsymbol{k_n},\boldsymbol{v_n}$ by linear projection
  for $i\leftarrow 1$ **to** T **do**
    for $j\leftarrow 1$ **to** T **do**
      $\mu_{ij},\sigma_{ij}=MLP(\boldsymbol{q_n^i},\boldsymbol{k_n^j})$
      Compute $\mathcal{L}_{nll}$ using Eq. (10)
      Sample α_{ij} using Eq. (4)
    end for
    Compute z_i using Eq. (3)
  end for
  $\boldsymbol{Z_{norm}}=LN(z_1,...,z_T)$ // Layer normalization
  $\boldsymbol{Z_{out}}=FFN(\boldsymbol{Z_{norm}})+\boldsymbol{Z_{norm}}$ // Feed forward
  $p(y_n|\boldsymbol{X_n},\Theta,\boldsymbol{\alpha_n})=softmax(\boldsymbol{Z_{out}})$
  Compute $\mathcal{I}(\boldsymbol{\alpha_n},y_n|\boldsymbol{X_n},\Theta)$ using Eq. (9)
  Compute $\mathcal{L}_{ma}$ and $\mathcal{L}_{mi}$ using Eq. (11)
  Update Θ by minimizing $\mathcal{L}_{ma}$ or $\mathcal{L}_{mi}$
end for
return Θ

In this NLL loss function, the mean of the Gaussian distribution is set as the scaled dot-product attention. During training, this NLL loss is jointly minimized with the classification cross-entropy loss.

Given the estimated epistemic uncertainty, we now introduce our uncertainty-guided training procedure. The uncertainty implies that the value of each sample for our model varies. Our model is unfamiliar with the data that have high epistemic uncertainty. This is an important information to improve the model. To guide the model training and inference, we propose a majority model $p_{ma}(y|\boldsymbol{X})$ and a minority model $p_{mi}(y|\boldsymbol{X})$, which respectively focus on the low-uncertainty data and high-uncertainty data. For majority model, data with low uncertainty are assigned higher weights during the training. Actually, the majority part of data belong to low-uncertainty group. On the other hand, data with high epistemic uncertainty are handled by the minority model, which assign higher weights to data with high epistemic uncertainty. The uncertainty-guided loss functions of both models are defined as:

$$\begin{aligned}\mathcal{L}_{ma} &= (1-\frac{exp(\mathcal{I}[y',\boldsymbol{\alpha}|\boldsymbol{X_n},\Theta^*])}{\sum_{n=1}^{N}exp(\mathcal{I}[y',\boldsymbol{\alpha}|\boldsymbol{X_n},\Theta^*])})\mathcal{L}_n+w\mathcal{L}_{nll} \\ \mathcal{L}_{mi} &= (1+\frac{exp(\mathcal{I}[y',\boldsymbol{\alpha}|\boldsymbol{X_n},\Theta^*])}{\sum_{n=1}^{N}exp(\mathcal{I}[y',\boldsymbol{\alpha}|\boldsymbol{X_n},\Theta^*])})\mathcal{L}_n+w\mathcal{L}_{nll}\end{aligned} \tag{11}$$

where $\mathcal{L}_n$ is the standard cross-entropy loss for classification of nth training sample and w is the weight for the negative log-likelihood loss, which is a hyperparameter. To prevent the negative log-likelihood loss from dominating

the training, we set $w = 0.15$ through experiments. The total loss is the weighted sum of the uncertainty-weighted cross-entropy loss and the negative log-likelihood loss. Using the majority loss $\mathcal{L}_{ma}$ and minority loss $\mathcal{L}_{mi}$, we train the majority model and minority model separately, then we combine these two models to cover all the data to utilize the epistemic uncertainty. The whole training process is summarized as Algorithm 1.

3.2.5 UGPT inference

Given the trained majority and minority probabilistic Transformers, complex action recognition by each model is performed using Eq. (6). To take advantage of both the majority model and the minority model, we combine them during the inference. As shown in Fig. 3, the two models are dynamically combined based on the input uncertainty to make the prediction. Specifically, the prediction is the weighted sum of the two predictions:

$$\begin{aligned} P(y|\boldsymbol{X'},\Theta^*) = w_{ma}(\boldsymbol{X'})p_{ma}(y|\boldsymbol{X'},\Theta^*) \\ +w_{mi}(\boldsymbol{X'})p_{mi}(y|\boldsymbol{X'},\Theta^*) \end{aligned} \tag{12}$$

where w_{ma} and w_{mi} are the weights of majority model and minority model respectively. These two weights are dependent of the inputs and are adaptively computed based on the epistemic uncertainty as follow:

$$\begin{aligned} w_{ma}(\boldsymbol{X'}) &= \sigma(k\frac{\mathcal{I}_{max} - \mathcal{I}[y',\boldsymbol{\alpha}|\boldsymbol{X'},\Theta^*]}{\mathcal{I}_{max} - \mathcal{I}_{min}} + b) \\ w_{mi}(\boldsymbol{X'}) &= 1 - w_{ma}(\boldsymbol{X'}) \end{aligned} \tag{13}$$

where $\mathcal{I}_{max}$ and $\mathcal{I}_{min}$ are the maximum uncertainty and minimum epistemic among samples respectively, $\mathcal{I}[y',\boldsymbol{\alpha}|\boldsymbol{X'},\Theta^*]$ is the epistemic uncertainty of input $\boldsymbol{X'}$ computed by Eq. (9), $\sigma()$ is the sigmoid function, and k, b are the learnable parameters. From the weights equations, high uncertainty input tends to have larger w_{mi} to rely on the power of minority model for prediction and vice verse. The whole inference process is summarized as Algorithm 2.

Algorithm 2 UGPT Inference

Input: $\mathcal{D}' = \{\boldsymbol{X'_n}\}$: testing data
Output: $\{y'\}$: predicted labels

for $\boldsymbol{X'_n}$ in $\mathcal{D}'$ **do**
  Compute $\boldsymbol{q_n}, \boldsymbol{k_n}, \boldsymbol{v_n}$ by linear projection
  for $i \leftarrow 1$ **to** T **do**
    for $j \leftarrow 1$ **to** T **do**
      $\mu_{ij}, \sigma_{ij} = MLP(\boldsymbol{q_n^i}, \boldsymbol{k_n^j})$
      for $k \leftarrow 1$ **to** K **do**
        Sample α_{ij}^k using Eq. (4)
      end for
    end for
    Compute z_i^k using Eq. (3)
  end for
  $\boldsymbol{Z_{norm}} = LN(z_1, ..., z_T)$ // Layer normalization
  $\boldsymbol{Z_{out}} = FFN(\boldsymbol{Z_{norm}}) + \boldsymbol{Z_{norm}}$ // Feed forward
  $p(y'_n|\boldsymbol{X'_n}, \Theta^*, \boldsymbol{\alpha_n}) = softmax(\boldsymbol{Z_{out}})$
  Compute $\mathcal{I}(\boldsymbol{\alpha_n}, y'_n|\boldsymbol{X'_n}, \Theta^*)$ using Eq. (9)
  Compute w_{ma}, w_{mi} using Eq. (13)
  Compute $P(y'_n|\boldsymbol{X'_n}, \Theta^*)$ using Eq. (12)
  Solve y'_n by Eq. (6)
end for
return $\{y'\}$

4. Experiments

In this part, we first introduce the benchmark datasets in Sec. 4.1. Then the implementation and training details are provided in Sec. 4.2. The experiment results and comparison are discussed in Sec. 4.3. We analyze the uncertainty in Sec. 4.4. Finally, we give ablation studies in Sec. 4.5.

4.1. Datasets

Breakfast Actions [18] is a dataset of 10 cooking activities in multiple real-life kitchens. There are totally over 77 hours of 1712 videos with an average of 2.3 minutes per video. We choose 1357 videos for training and 335 for testing. All the complex actions are composed of 48 classes of atomic actions as well as background. The dataset also provides temporal annotations for atomic actions. We evaluate both the accuracy and mean average precision.

MultiTHUMOS is an extension of the THUMOS dataset [15] for human activity recognition from unconstrained internet videos. There are totally 413 videos (30h) in 65 classes with 1.5 labels per frame. Each video has 10.5 action categories on average. The original purpose of this dataset is for action detection. But due to the temporal structure of its actions, this dataset can also be used for complex action recognition.

Charades [28] is a large-scale dataset for action classification. It contains 9848 annotated videos of 157 action classes with 27847 video descriptions and 66500 temporally localized intervals. Each complex action (one video) is about 30 seconds long and contains 6 atomic actions on average. We split the dataset into 7985 videos for training and 1863 videos for testing following the metrics in [28].

4.2. Implementation and training details

We implemented our method in PyTorch 1.6.0 [22]. Given video sequences with variant lengths, we first trained the atomic action localization module in an unsupervised manner [3]. The atomic action localization module is fine-tuned during the training of complex action recognition. For the feature extraction, we use a CNN-based backbone to ex-

Method	Activities (Acc. %)	Actions (mAP %)
I3D [1]	64.31	47.71
3D ResNet-50 [10]	66.73	53.27
ActionVLAD [9]	65.48	60.20
Timeception [14]	71.25	59.64
VideoGraph [13]	69.45	63.14
GHRM [50]	75.49	65.86
UGPT (ours)	**77.79**	**67.82**

Table 1. **Experiment results on Breakfast Actions.** All methods use the same Kinetics [1] pre-trained I3D backbone. Our proposed UGBT achieves the state-of-the-art performance.

tract the feature of each segment. The backbones we studied include I3D [1], 2D ResNet [11], 3D ResNet [10]. All the backbones are pretrained on Kinetics [1] dataset. For the probabilistic self-attention layers, we project each input to 5 heads for the multi-head attention mechanism. We set the number of self-attention layers as 6 and we show the impact of the number of layers in Sec. 4.5. The output embeddings of the final self-attention layer are fed into a two-layer linear classifier to make the final prediction. We set the batch size as 16 and SGD with a learning rate of 0.1.

4.3. Experiment results and comparison

The experiment results on Breakfast Actions are shown in Tab. 1. Compared with other methods using the I3D backbone pretrained on Kinetics [1], our UGBT achieves 77.79% accuracy for complex action recognition, which outperforms the SOTA method GHRM [50] by 2.30%. The experiment results on MultiTHUMOS dataset are shown in Tab. 2. Our method achieves 81.42% mAP using the I3D backbone, which outperforms the SOTA method by 1.53%. The experiment results on Chardes dataset are shown in Tab. 3. Using the same backbones, our method achieve the SOTA performance.

Method	mAP (%)
I3D [1]	72.53
Timeception [14]	74.79
GHRM [50]	79.89
UGPT (ours)	**81.42**

Table 2. **Experiment results on MultiTHUMOS.** Our proposed UGPT outperforms the baseline method I3D [1] by 8.89% and the state-of-the-art GHRM [50] by 1.53%.

4.4. Analysis of epistemic uncertainty for complex action recognition

The recognition accuracy of each complex action on Breakfast Actions is shown in Fig. 4. We rank the complex actions by their epistemic uncertainties from left to right in a ascending order. The complex actions with lower epistemic uncertainty have higher recognition accuracy and vice versa. By guiding the training and inference using epistemic uncertainty. The recognition of complex actions

Ours	Method	Modality	mAP (%)
	Two-stream [29]	RGB + Flow	18.6
	Two-stream + LSTM [29]	RGB + Flow	17.8
	ActionVLAD [9]	RGB + iDT	21.0
	Temporal Fields [27]	RGB + Flow	22.4
	TRN [49]	RGB + Flow	25.2
	ResNet-152 [11]	RGB	22.8
	ResNet-152 + TC [14]	RGB	31.6
✓	ResNet-152 [11] + UGPT	RGB	**35.7**
	I3D [1]	RGB	32.9
	I3D + TC [14]	RGB	37.2
	VideoGraph [13]	RGB	37.8
	GHRM [50]	RGB	38.3
✓	I3D + UGPT	RGB	**38.8**
	3D ResNet-101 + NL [38]	RGB + RP	37.5
	3D ResNet-101 + GCN [39]	RGB + RP	39.7
	3D ResNet-101 + TC [14]	RGB	41.1
✓	3D ResNet-101 + UGPT	RGB	**42.4**

Table 3. **Experiment results on Charades.** Using different CNN-based backbones, our proposed uncertainty-guided probabilistic Transformer achieves state-of-the-art performance on Charades.

with high epistemic uncertainty are significantly improved by the minority model. For example, the performance of actions like "scrambled egg" and "making milk" are improved from 59.0% to 63.2% and 63.4% to 69.5 respectively. Larger epistemic uncertainty implies that the current model is not familiar with the data and not confident about the prediction. According to Eq. (11), these high-uncertainty samples are assigned larger weights during the training. Then these high-uncertainty samples are well addressed and thus the performance improves. On the other hand, the majority model assigns higher weights to samples with low epistemic uncertainties. Also, we notice that

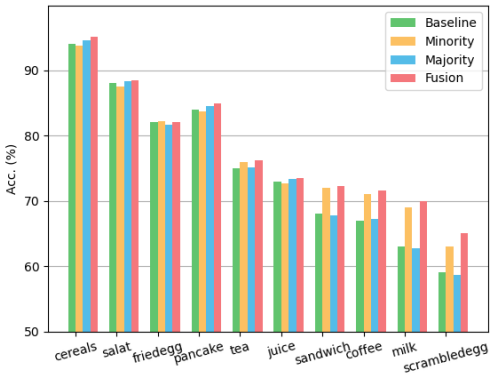

Figure 4. **Acc. of each complex action in Breakfast Actions** [18]. Actions are ranked by epistemic uncertainty in a ascending order from left to right. The minority model improves the recognition of high-uncertainty samples significantly compared with the baseline model. The majority model improves the recognition of low-uncertainty samples. The combination of these two models (fusion model) gives the best performance.

the remaining samples of each model keep robust performance while adopting the uncertainty guiding strategy. By combining the majority model and minority model during the inference based on the epistemic uncertainty of the test sample, the fusion model achieves better performance than either majority model or minority model.

4.5. Ablation study and analysis

Impact of the number of self-attention layers. In this part, we show the performance of our proposed UGPT by stacking different numbers of probabilistic self-attention layers. We set the number of layers from 1 to 6. The experiment results on Breakfast Actions and MultiTHUMOS are shown in Fig. 5. The performance reaches its peak when we set the number of layers to 6.

Is atomic action localization necessary? Our model includes an unsupervised atomic localization module, is it necessary or helpful? We conducted additional experiments by directly feeding the features to the UGPT without the atomic action localization. For Breakfast Actions, the accuracy reached 78.56% without localization compared to 77.79% with localization. Although using the raw features gives a better performance, we still claim it is necessary to include the atomic action localization module since the improvement is marginal and we can have a better understanding of complex actions about their temporal contents and structures. Also, by fine-tuning the atomic action localization module during the joint training, the localization of atomic actions also shows improvement from 70.2% to 70.4%.

Using less training data. In many cases, the training data is insufficient, which limits the performance of the model. As our probabilistic Transformer can capture the stochasticity of both the model and the data, it can still be robust with insufficient training data. To demonstrate our claim, we reduce the amount of training data for our model from 100% to 20% and perform the experiment in the same settings. The experiment results are shown in Tab. 4. Compared with the Transformer in deterministic mode, our proposed UGPT perform much better with limited training data.

Portion of data	100%	80%	60%	40%	20%
Deterministic	74.22	69.52	61.05	50.91	37.09
UGPT	77.79	74.67	68.83	57.64	46.70

Table 4. Comparison of deterministic Transformer and UGPT using **less training data** on Breakfast Actions. We reduce the training data from 100% to 20%. The UGPT is more robust under insufficient data compared with deterministic setting.

Effectiveness of NLL loss. During the training of the probabilistic Transformer, we also use a negative log-likelihood loss to train the probabilistic attention besides the cross-entropy loss. The proposed NLL loss function is multiplied by a weight w in the total loss function. To make it effective for capturing dependencies and avoid it dominating the training. We use different w as shown in Tab. 5. The experiment results show that large w make the NLL dominate the training and thus degenerates the performance. To get a balance, we set $w = 0.15$ for the NLL loss in this work.

w	0	0.05	0.10	0.15	0.20	0.30
acc. (%)	76.69	76.98	77.24	77.79	70.52	68.38

Table 5. **Effectiveness of NLL loss.** The acc. in the table is obtained on Breakfast Actions. We pick $w = 0.15$ for our model in order to leverage the NLL constrain and avoid the NLL loss from dominating the training.

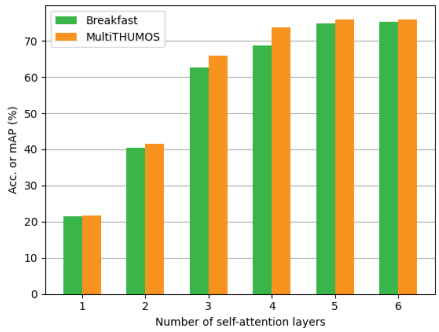

Figure 5. **Impact of the number of probabilistic self-attention layers.** By stacking the self-attention layers, we found the performance reaches peak when we set the number of of layers as 6.

5. Conclusion and Future Work

Conclusion. In this paper, we introduce the Uncertainty-Guided Probabilistic Transformer (UGPT) for complex action recognition. We model the attention scores as Gaussian random variables in order to capture the stochasticity and uncertainty in the data and prediction. We also propose a novel training and inference strategy guided by the epistemic uncertainty. Our method achieves state-of-the-art performance on Breakfast Actions, MultiTHUMOS, and Charades dataset. Utilizing the probabilistic method and modeling the uncertainty, our model is also robust when training data is limited.

Future work. In this work, we use sampling-based method to estimate the epistemic uncertainty. There are also other uncertainty quantification methods such as the methods introduced in Sec. 2. We may evaluate different uncertainty quantification methods in the future. Also, the proposed uncertainty-guided training and inference strategy may be evaluated on other computer vision tasks.

Acknowledgment: The work described in this paper is supported in part by the U.S. National Science Foundation award CNS 1629856, the DARPA grant FA8750-17-2-0132, and by the Cognitive Immersive Systems Laboratory (CISL), a collaboration between IBM and RPI.

References

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 1, 7

[2] Jie Chang, Zhonghao Lan, Changmao Cheng, and Yichen Wei. Data uncertainty learning in face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5710–5719, 2020. 2

[3] Min-Hung Chen, Baopu Li, Yingze Bao, Ghassan AlRegib, and Zsolt Kira. Action Segmentation with Joint Self-Supervised Temporal Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9454–9463, 2020. 3, 6

[4] Jen-Tzung Chien and Yuan-Chu Ku. Bayesian recurrent neural network for language modeling. *IEEE transactions on neural networks and learning systems*, 27(2):361–374, 2015. 2

[5] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 1

[6] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 6202–6211, 2019. 1

[7] 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

[8] Kirill Gavrilyuk, Ryan Sanford, Mehrsan Javan, and Cees G. M. Snoek. Actor-transformers for group activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 839–848, 2020. 1

[9] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 971–980, 2017. 7

[10] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018. 7

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3, 7

[12] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 1

[13] Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Videograph: Recognizing minutes-long human activities in videos. *arXiv preprint arXiv:1905.05143*, 2019. 2, 7

[14] Noureldien Hussein, Efstratios Gavves, and Arnold W. M. Smeulders. Timeception for complex action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 254–263, 2019. 2, 7

[15] Haroon Idrees, Amir R. Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The THUMOS challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155:1–23, 2017. 6

[16] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *arXiv preprint arXiv:1703.04977*, 2017. 2

[17] Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. *Advances in neural information processing systems*, 28:2575–2583, 2015. 4

[18] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 780–787, 2014. 1, 2, 6, 7

[19] Fang Liu, Xiangmin Xu, Tong Zhang, Kailing Guo, and Lin Wang. Exploring privileged information from simple actions for complex action recognition. *Neurocomputing*, 380:236–245, 2020. 2

[20] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2906–2917, 2021. 1

[21] Juan Carlos Niebles, Chih-Wei Chen, and Li Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *European conference on computer vision*, pages 392–405. Springer, 2010. 1, 2

[22] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 6

[23] Ronald Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010. 1

[24] Isidoros Rodomagoulakis, Nikolaos Kardaris, Vassilis Pitsikalis, E Mavroudi, Athanasios Katsamanis, Antigoni Tsiami, and Petros Maragos. Multimodal human action recognition in assistive human-robot interaction. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2702–2706. IEEE, 2016. 1

[25] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016. 4

[26] Fadime Sener and Angela Yao. Unsupervised learning and segmentation of complex activities from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8368–8376, 2018. 2

[27] Gunnar A Sigurdsson, Santosh Divvala, Ali Farhadi, and Abhinav Gupta. Asynchronous temporal fields for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 585–594, 2017. 7

[28] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in

homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pages 510–526. Springer, 2016. 6

[29] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. 1, 7

[30] Deepika Singh, Erinc Merdivan, Ismini Psychoula, Johannes Kropf, Sten Hanke, Matthieu Geist, and Andreas Holzinger. Human activity recognition using recurrent neural networks. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 267–274. Springer, 2017. 1

[31] Lewis Smith and Yarin Gal. Understanding measures of uncertainty for adversarial example detection. *arXiv preprint arXiv:1803.08533*, 2018. 2

[32] Mahesh Subedar, Ranganath Krishnan, Paulo Lopez Meyer, Omesh Tickoo, and Jonathan Huang. Uncertainty-aware audiovisual activity recognition using deep bayesian variational inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6301–6310, 2019. 2

[33] Eran Swears, Anthony Hoogs, Qiang Ji, and Kim Boyer. Complex activity recognition using granger constrained dbn (gcdbn) in sports and surveillance video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 788–795, 2014. 1

[34] Kevin Tang, Li Fei-Fei, and Daphne Koller. Learning latent temporal structure for complex event detection. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1250–1257. IEEE, 2012. 2

[35] Dustin Tran, Mike Dusenberry, Mark van der Wilk, and Danijar Hafner. Bayesian layers: A module for neural network uncertainty. *Advances in Neural Information Processing Systems*, 32:14660–14672, 2019. 2

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 1, 3

[37] Limin Wang, Yu Qiao, and Xiaoou Tang. Latent hierarchical model of temporal structure for complex activity classification. *IEEE Transactions on Image Processing*, 23(2):810–822, 2013. 2

[38] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 7

[39] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *Proceedings of the European conference on computer vision (ECCV)*, pages 399–417, 2018. 7

[40] Zhenyu Wang, Yali Li, Ye Guo, Lu Fang, and Shengjin Wang. Data-uncertainty guided multi-phase learning for semi-supervised object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4568–4577, 2021. 2

[41] Boyang Xue, Jianwei Yu, Junhao Xu, Shansong Liu, Shoukang Hu, Zi Ye, Mengzhe Geng, Xunying Liu, and Helen Meng. Bayesian transformer language models for speech recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7378–7382. IEEE, 2021. 2

[42] An Yan, Yali Wang, Zhifeng Li, and Yu Qiao. PA3D: Pose-action 3D machine for video recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7922–7931, 2019. 1

[43] Fan Yang, Qiang Zhai, Xin Li, Rui Huang, Ao Luo, Hong Cheng, and Deng-Ping Fan. Uncertainty-guided transformer reasoning for camouflaged object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4146–4155, 2021. 2

[44] Serena Yeung, Olga Russakovsky, Ning Jin, Mykhaylo Andriluka, Greg Mori, and Li Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *International Journal of Computer Vision*, 126(2-4):375–389, 2018. 1

[45] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pages 7354–7363. PMLR, 2019. 1

[46] Shujian Zhang, Xinjie Fan, Bo Chen, and Mingyuan Zhou. Bayesian attention belief networks. *arXiv preprint arXiv:2106.05251*, 2021. 2

[47] Yongmian Zhang, Yifan Zhang, Eran Swears, Natalia Larios, Ziheng Wang, and Qiang Ji. Modeling temporal interactions with interval temporal bayesian networks for complex activity recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(10):2468–2483, 2013. 1

[48] Rui Zhao, Kang Wang, Hui Su, and Qiang Ji. Bayesian graph convolution lstm for skeleton based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6882–6892, 2019. 2

[49] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 803–818, 2018. 7

[50] Jiaming Zhou, Kun-Yu Lin, Haoxin Li, and Wei-Shi Zheng. Graph-based high-order relation modeling for long-term action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8984–8993, 2021. 2, 7