

# Uncertainty Quantification in Extreme Classification

Jyun-Yu Jiang  
jyunyu.jiang@gmail.com  
Amazon Search  
Palo Alto, California, USA

Wei-Cheng Chang  
weicheng.cmu@gmail.com  
Amazon Search  
Palo Alto, California, USA

Jiong Zhang  
zhangjiong724@gmail.com  
Amazon Search  
Palo Alto, California, USA

Cho-Jui Hsieh  
chohsieh@cs.ucla.edu  
University of California, Los Angeles  
Los Angeles, California, USA

Hsiang-Fu Yu  
rofu.yu@gmail.com  
Amazon Search  
Palo Alto, California, USA

## ABSTRACT

Uncertainty quantification is one of the most crucial tasks to obtain trustworthy and reliable machine learning models for decision making. However, most research in this domain has only focused on problems with small label spaces and ignored eXtreme Multi-label Classification (XMC), which is an essential task in the era of big data for web-scale machine learning applications. Moreover, enormous label spaces could also lead to noisy retrieval results and intractable computational challenges for uncertainty quantification. In this paper, we aim to investigate general uncertainty quantification approaches for tree-based XMC models with a probabilistic ensemble-based framework. In particular, we analyze label-level and instance-level uncertainty in XMC, and propose a general approximation framework based on beam search to efficiently estimate the uncertainty with a theoretical guarantee under long-tail XMC predictions. Empirical studies on six large-scale real-world datasets show that our framework not only outperforms single models in predictive performance, but also can serve as strong uncertainty-based baselines for label misclassification and out-of-distribution detection, with significant speedup. Besides, our framework can further yield better state-of-the-art results based on deep XMC models with uncertainty quantification.

## CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Information systems** → **Information retrieval**.

## KEYWORDS

extreme multi-label classification, uncertainty quantification, Bayesian ensemble

### ACM Reference Format:

Jyun-Yu Jiang, Wei-Cheng Chang, Jiong Zhang, Cho-Jui Hsieh, and Hsiang-Fu Yu. 2023. Uncertainty Quantification in Extreme Classification. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*, July 23–27, 2023, Taipei, Taiwan. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3539618.3591780>



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs International 4.0 License.

SIGIR '23, July 23–27, 2023, Taipei, Taiwan

© 2023 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-9408-6/23/07.

<https://doi.org/10.1145/3539618.3591780>

## 1 INTRODUCTION

Extreme multi-label classification (XMC), or extreme multi-label learning, aims to find the relevant labels for a data input from an enormous label space. With increasingly growing information in the era of big data, XMC has become more and more important, and has been widely applied to diverse fields and various real-world applications, such as advertising [39], product search [9], webpage annotation [53], document retrieval [7], and hashtag recommendation [14]. However, for domains with potential high risks and costs from mistakes, such as public health, biology, and medicine, it is essentially crucial to model the predictive uncertainty for their downstream XMC applications like food classification [57], microbiome identification [49], and medical diagnosis [2, 5]. As a result, there is a need to study uncertainty quantification for XMC models, which is still a virgin territory in this field.

To estimate predictive uncertainty, Bayesian and probabilistic models [22] are inherently applicable because variance can intrinsically be viewed as an uncertainty measurement. However, although Bayesian approaches are mathematically grounded to model uncertainty, their computational costs are usually exorbitant for large-scale data. To address this issue, the most popular solution is to approximate Bayesian inference by sampling models as an ensemble [19]. Accordingly, ensemble-based Bayesian approximation has been applied to analyze the uncertainty in neural networks [15, 20], gradient boosting based on decision trees [38], autoregressive structured prediction [36], and random forests [42]. Unfortunately, none of the existing studies has studied uncertainty quantification for XMC models.

Everything happens for a reason, and so the underexplored area of modeling uncertainty for XMC does. There are several challenges when it comes to uncertainty quantification for XMC models. First, different from conventional classification models, XMC models usually focus on deriving a small subset of relevant labels from the enormous label space. This is also why XMC models are usually evaluated by ranking metrics. In other words, for XMC models, we not only should consider how confident a model is for the entire input, but also need to model the uncertainty for each individual label. However, most of the existing uncertainty quantification studies only concentrate on instance-level uncertainty [1]. The enormous number of labels could also result in computational difficulty. Second, XMC models usually take extremely sparse input features [9, 52] with distillation [54]. Moreover, many XMC models [39, 54] conduct convex optimization for better computational

efficiency, which are insensitive to initialization [41]. As a result, existing uncertainty quantification approaches that manipulate model weights, such as MC Dropout [20] and Deep Ensemble [29], could be ineffective.

To address these issues, in this work, we investigate ensemble-based uncertainty quantification for XMC models. We first present the concept of modeling *label-level* and *instance-level* uncertainty. To tackle the efficiency issue due to enormous labels, we propose a general framework to approximate uncertainty measurements by beam search with a theoretical guarantee under long-tail probability distributions. Our contributions can be summarized as:

- We are the pioneer of uncertainty quantification for XMC models. Especially, we broaden the scope by simultaneously modeling *label-level* and *instance-level* uncertainty, which is an essence of XMC tasks.
- We propose an efficient and general framework to approximate uncertainty in XMC. With the observation of long-tail distributions in predictions, beam search on tree-based XMC models can further mitigate the computational efficiency with mathematical guarantees on estimated uncertainty.
- We conduct experiments on six benchmark XMC datasets. Our approaches can not only obtain satisfactory predictive performance, but also appropriately estimate both *label-level* and *instance-level* uncertainty. Besides, using XR-TRANSFORMER as deep base models, our framework can obtain better state-of-the-art results.

## 2 PRELIMINARIES

In this section, we introduce background and essential notions about extreme multi-label classification (XMC) and uncertainty quantification.

### 2.1 Ensemble-based Uncertainty Quantification

In this work, we focus on ensemble-based uncertainty quantification using Bayesian ensembles [19], which learns the ensemble of multiple individual models. The model parameters  $\theta$  are considered as random variables from a posterior distribution  $p(\theta | \mathcal{D})$ , which can be computed by the Bayes' rule as:

$$p(\theta | \mathcal{D}) = \frac{p(\mathcal{D} | \theta)p(\theta)}{p(\mathcal{D})}$$

where  $\mathcal{D} = \{x_i, y_i\}$  is the training dataset, and  $p(\mathcal{D})$  is the prior data distribution. More precisely, each set of model parameters  $\theta$  is a sample from the posterior  $p(\theta | \mathcal{D})$  to demonstrate a hypothesis [6] learned from the observations presented by the training data  $\mathcal{D}$ .

Since the exact Bayesian inference could be intractable, a conventional approach is to consider an approximated distribution  $q(\theta)$  and mimic the true posterior  $p(\theta | \mathcal{D})$  [6]. Specifically, exploiting ensemble models is one of the most popular approximation methods [12]. Suppose we have an ensemble of  $M$  probabilistic models  $\{\Pr(y | x; \theta^{(m)})\}_{m=1}^M$  sampled from the posterior  $p(\theta | \mathcal{D})$ . The *predictive posterior*  $\Pr(y | x, \mathcal{D})$  for inference approximation based on the ensemble can be estimated by computing the expectation

over the ensemble models as:

$$\begin{aligned} \Pr(y | x, \mathcal{D}) &= \int_{\theta} p(y | x; \theta)p(\theta | \mathcal{D})d\theta \\ &\approx \mathbb{E}_{q(\theta)}[\Pr(y | x; \theta)] \approx \frac{1}{M} \sum_{m=1}^M \Pr(y | x; \theta^{(m)}), \end{aligned}$$

where  $\theta^{(m)} \sim q(\theta) \approx p(\theta | \mathcal{D})$  represents the model parameters of each individual model in the ensemble.

**Uncertainty Quantification via Entropy.** For a probabilistic model and its outputs, entropy is a native way to estimate the uncertainty [48]. Given the predictive posterior  $\Pr(y | x, \mathcal{D})$ , the overall uncertainty, or so-called *total uncertainty* [15], can be estimated as:

$$\begin{aligned} \mathcal{H}[\Pr(y | x, \mathcal{D})] &= \mathbb{E}_{p(y|x, \mathcal{D})}[-\ln \Pr(y | x, \mathcal{D})] \\ &\approx \mathcal{H}\left[\frac{1}{M} \sum_{m=1}^M \Pr(y | x; \theta^{(m)})\right]. \end{aligned} \quad (1)$$

However, *total uncertainty* incorporates both epistemic (*knowledge*) uncertainty and aleatoric (*data*) uncertainty [15, 27]. Previous studies also demonstrate that *knowledge uncertainty* could be more beneficial for downstream applications, such as active learning [27] and out-of-distribution detection [11]. To compute the *knowledge uncertainty*, we can decompose the *total uncertainty* by deriving the mutual information between the model parameters  $\theta$  and the prediction  $y$  [15] as:

$$\begin{aligned} \underbrace{\mathcal{I}[y, \theta | x, \mathcal{D}]}_{\text{Knowledge Uncertainty}} &= \underbrace{\mathcal{H}[\Pr(y | x, \mathcal{D})]}_{\text{Total Uncertainty}} - \underbrace{\mathbb{E}_{p(\theta|\mathcal{D})}[\mathcal{H}[\Pr(y | x; \theta)]]}_{\text{Expected Data Uncertainty}} \\ &\approx \mathcal{H}\left[\frac{1}{M} \sum_{m=1}^M \Pr(y|x; \theta^{(m)})\right] - \frac{1}{M} \sum_{m=1}^M \mathcal{H}[\Pr(y|x; \theta^{(m)})]. \end{aligned} \quad (2)$$

**Uncertainty Quantification via Variation.** If we treat the inference process as a regression problem to derive continuous probabilities, the variation over predicted probabilities of individual models in the ensemble can be considered as another direction to estimate the uncertainty [43]. Specifically, the uncertainty can be estimated by computing the *probability variance* as:

$$\begin{aligned} \mathbb{V}_{p(y|x, \mathcal{D})}[\Pr(y | x; \theta)] &\approx \mathbb{V}_{q(\theta)}[\Pr(y | x; \theta)] \\ &\approx \frac{1}{M} \sum_{m=1}^M [\Pr(y | x; \theta^{(m)}) - \mu]^2, \end{aligned} \quad (3)$$

where  $\mu = \frac{1}{M} \sum_{m=1}^M \Pr(y | x; \theta^{(m)})$  is the mean predicted probability of all individual models in the ensemble.

In this work, we examine three uncertainty approaches in our experiments, including *total uncertainty* (TU), *knowledge uncertainty* (KU), and *probability variance* (PV).

## 2.2 eXtreme Multi-label Classification (XMC)

Given a training dataset  $\mathcal{D}$  of  $N$  instances, for an arbitrary testing instance  $\mathbf{x}$ , an XMC model aims to estimate a probability  $\Pr(y_\ell | \mathbf{x}, \mathcal{D})$  for each label  $\ell \in \mathcal{L}$  in an extreme label space  $\mathcal{L}$  with  $L$  labels.

**Training with Negative Example Selection.** With the extreme label space, it is time-consuming for models to be trained with all negative examples. As a result, most of the state-of-the-art XMC models select appropriate negative examples during optimization. For example, Parabel [39] and PECOS [54] consider teacher forcing negatives [30] and matcher-aware negatives for better performance. For simplicity, we denote the adjusted training dataset with negative example selection as  $\mathcal{D}'$  so that the targets of those XMC models become  $\Pr(y_\ell | \mathbf{x}, \mathcal{D}') \approx \Pr(y_\ell | \mathbf{x}, \mathcal{D})$ .

**Hierarchical Label Tree and Semantic Indexing.** Due to the enormous labels, it can be inappropriate to have linear inference time complexity to the number of labels. To achieve acceptable efficiency, one of the most prominent methods is to partition the enormous label space into a hierarchical label tree [26, 39, 54]. Specifically, top-down  $B$ -ary clustering can recursively construct a depth- $d$  hierarchical label tree with label representations like instance feature aggregation and additional label features. The cluster numbers  $K_t$  are  $B^t$  and  $L$  for the first  $d-1$  layers and the last  $d$ -th layer. The clustering assignment at each layer  $t$  can then be represented by an indexing matrix  $C^{(t)} \in \{0, 1\}^{K_t \times K_{t-1}}$  as:

$$C_{tk}^{(t)} = \begin{cases} 1, & \text{if } k = c_{\ell k}^{(t)} \\ 0, & \text{otherwise} \end{cases},$$

where  $c_{\ell k}^{(t)}$  is the corresponding cluster at the layer  $t-1$  for a cluster (or a label when  $t=d$ ) at the layer  $t$  in the hierarchical label tree.

With the hierarchical label tree and the indexing matrix  $C^{(t)}$ , tree-based XMC models can conduct semantic indexing with beam search [28] for efficient approximated inference in a sub-linear time. More precisely, for every clustering layer  $t-1$ , tree-based models estimate  $\Pr(y_\ell^t | \mathbf{x}, \mathcal{D}'^{(t)})$  and encode top- $b$  clusters into  $Y^{(t-1)} \in \{0, 1\}^{K_{t-1}}$ , where  $\mathcal{D}'^{(t)}$  is the induced training dataset for clusters at the layer  $t$ ;  $\sum Y^{(t-1)}[i] = b$ . Models can then focus on limited clusters or labels indicated by  $Y^{(t-1)} C^t \in \{0, 1\}^{K_t}$  for inference at the clustering layer  $t$ . Finally,  $\Pr(y_\ell | \mathbf{x}, \mathcal{D}')$  can then be computed as:

$$\Pr(y_\ell | \mathbf{x}, \mathcal{D}') = \Pr(y_\ell^d | \mathbf{x}, \mathcal{D}'^{(d)}) \times \Pr(c_\ell^d | \mathbf{x}, \mathcal{D}'^{(d-1)}),$$

where  $\Pr(c_\ell^d | \mathbf{x}, \mathcal{D}'^{(d-1)})$  can be further recursively derived through clustering layers. In this paper, we denote a tree-based XMC model learn a parameter set  $\theta$  from the dataset to recursively compute the predictions as:

$$\Pr(y_\ell | \mathbf{x}; \theta) = \Pr(y_\ell^d | \mathbf{x}, c_\ell^d; \theta) \times \Pr(c_\ell^d | \mathbf{x}; \theta). \quad (4)$$

**Challenges in XMC Uncertainty Quantification.** Although the uncertainty measures introduced in Section 2.1 are tractable in most scenarios, a linear computational time over all enormous labels could still be unacceptable. Besides, uncertainty for XMC models can be examined in two levels, including (1) *label-level* for the predicted probability  $\Pr(y_\ell | \mathbf{x}, \mathcal{D})$  of each single label  $\ell$ , and (2) *instance-level* for the entire instance  $\mathbf{x}$ . However, none of the

existing studies has addressed any of these two uncertainty types for XMC models.

## 3 XMC UNCERTAINTY QUANTIFICATION

In this section, we propose an efficient framework to approximate both label-level and instance-level uncertainty. We also suggest two approaches to generate model ensembles.

### 3.1 Uncertainty in XMC

Here we propose to model two different levels of uncertainty, including *label-level* and *instance-level* uncertainty. The former represents the confidence of the decision on each individual label, and the latter estimates how certain the model is for a given data instance.

**Label-level Uncertainty.** Label-level uncertainty quantification can be reduced to  $L$  subtasks of estimating uncertainty of binary classification for each label  $\ell \in \mathcal{L}$ . Specifically, given a model ensemble, we collect  $M$  predicted probabilities  $\{\Pr(y_\ell | \mathbf{x}; \theta^{(m)})\}_{m=1}^M$  for each label  $\ell$ , thereby estimating the uncertainty based on different approaches introduced in Equations (1), (2), and (3) in Section 2.1.

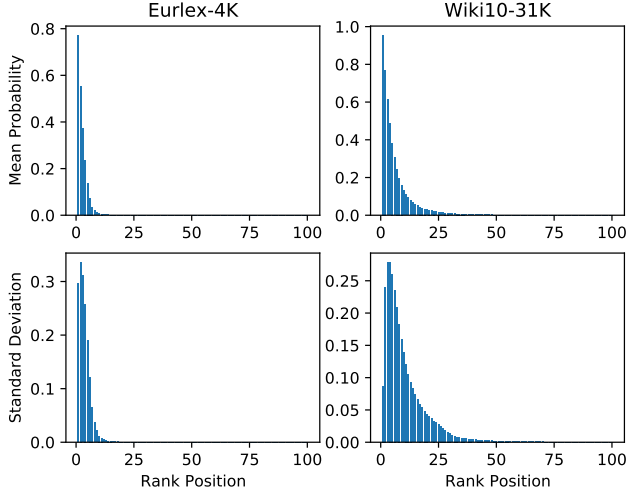
**Instance-level Uncertainty.** To estimate instance-level uncertainty, it is required to simultaneously consider the predictions over all labels  $\ell \in \mathcal{L}$  for an instance  $\mathbf{x}$ . As a pioneer of this direction in XMC, we borrow the idea of *JointEnergy* [31, 47] in the domain of conventional multi-label classification by adding up uncertainty metrics over all labels. Specifically, from the joint likelihood perspective, instance-level *Total Uncertainty*, which estimates the entropy for each label, can be decomposed into an explainable form under the assumption of conditional independence for  $p(y_\ell | \mathbf{x})$  as:

$$\begin{aligned} \sum_{\ell \in \mathcal{L}} \mathcal{H}[p(y_\ell)] &= \sum_{\ell \in \mathcal{L}} -\ln p(y_\ell | \mathbf{x}) \\ &= -\sum_{\ell \in \mathcal{L}} \ln p(\mathbf{x} | y_\ell) - \sum_{\ell \in \mathcal{L}} Z_\ell \\ &= -\ln \prod_{\ell \in \mathcal{L}} \frac{p(y_\ell | \mathbf{x}) p(\mathbf{x})}{p(y_\ell)} - \sum_{\ell \in \mathcal{L}} Z_\ell \\ &= -\ln p(y_1, \dots, y_L | \mathbf{x}) - L \ln p(\mathbf{x}) + \ln \prod_{\ell \in \mathcal{L}} p(y_\ell) - \sum_{\ell \in \mathcal{L}} Z_\ell \\ &= -\ln \frac{p(\mathbf{x} | y_1, \dots, y_L) \prod_{\ell \in \mathcal{L}} p(y_\ell)}{p(\mathbf{x})} - L \ln p(\mathbf{x}) \\ &\quad + \ln \prod_{\ell \in \mathcal{L}} p(y_\ell) - \sum_{\ell \in \mathcal{L}} Z_\ell \\ &= \underbrace{-\ln p(\mathbf{x} | y_1, \dots, y_L)}_{\text{Knowledge Uncertainty}} - \underbrace{(L-1) \ln p(\mathbf{x})}_{\text{Data Uncertainty}} - \sum_{\ell \in \mathcal{L}} Z_\ell, \end{aligned}$$

where  $Z_\ell = p(\mathbf{x} | y_\ell) / p(y_\ell | \mathbf{x})$  is the normalized density for each label  $\ell \in \mathcal{L}$ .

### 3.2 Uncertainty Approximation via Beam Search

Even though we now have tractable approaches to estimate both label-level uncertainty and instance-level uncertainty, their linear computational time could still be too slow to address XMC tasks that usually have millions of labels. In this work, we propose to leverage



**Figure 1: The mean and standard deviation of predicted probabilities over different rank positions on the testing data of Eurlex-4K and Wiki10-31K datasets.**

the power of long-tail probabilistic distributions and variances in XMC problems.

Figure 1 illustrates the statistics of predicted probabilities over different rank positions on the testing data of the Eurlex-4K and Wiki10-31K datasets. Both mean and standard deviation of predicted probabilities are long-tail over rank positions while most of the labels in the space are predicted with near-zero probabilities and variance. In other words, the label-level uncertainty of those labels and the corresponding components in the computations of instance-level uncertainty would be also close to zero. Indeed, instances in XMC tasks tend to have very few labels, so the phenomenon is also intuitive. Based on this observation, we propose to approximate uncertainty based on beam search and retrieving top-ranked but limited labels.

As mentioned in Section 2.2, beam search [28] is a convenient tool to efficiently derive the top-ranked labels. We also notice that the long-tail phenomenon on intermediate probabilities is also consistent through all layers in the search. Hence, for uncertainty approximation, we propose to only consider the predicted probabilities of top-ranked labels returned by beam search and assign zero probabilities for the remaining labels. The computational time of uncertainty can then be improved from linear to sub-linear in the number of labels. In Algorithm 1, we present the pseudo code of the detail process for uncertainty approximation via beam search for a testing instance  $\mathbf{x}$ .

### 3.3 Theoretical Analysis

In addition to the efficiency, the long-tail property also provides the theoretical guarantee on the accuracy of beam search results.

**Theorem 3.1.** *Suppose  $\Pr(y_\ell^t | \mathbf{x}; \theta)$  given by a tree-based model for each layer  $t$  is under a long tail distribution as shown in Figure 1. With a large enough beam size  $k^t$ , the average regret of beam search*

---

#### Algorithm 1 Uncertainty Approximation via Beam Search

---

**Input:** a model ensemble  $\{\Pr(y | \mathbf{x}; \theta^{(m)})\}_{m=1}^M$ ; indexing matrices  $\{C^{(t)}\}_{1 \leq t \leq d}$ ; the testing instance  $\mathbf{x}$ .

**Output:** Approximated Uncertainty Metrics  $\hat{U}$ .

**Hyper-parameters:** the beam width  $b$ ; the number of top-ranked labels  $k$ .

labelProb =  $\{\delta\}^{L \times M}$ , where  $\delta \approx 0$  is a small positive constant.

**for**  $m = 1$  **to**  $M$  **do**

Let  $A = \mathbf{1}^1$  denote label/cluster candidates over layers.

**for**  $t = 1$  **to**  $d$  **do**

$A = \text{Binarize}(C^{(t)\top} \times A)$

Calculate  $\Pr(y_\ell^t | \mathbf{x}; \theta^{(m)})$  for each candidate  $\ell$  in  $A$ .

$k_t = b$  **if**  $t < d$  **else**  $k$

$A = \mathbf{0}^{K_t}$

$A[\ell \text{ in top-}k_t \text{ candidates}] = 1$

**end for**

**for** top-ranked  $\ell$  **in**  $A$  **do**

labelProb[ $\ell, m$ ] =  $\Pr(y_\ell | \mathbf{x}; \theta^{(m)})$

**end for**

**end for**

Compute  $\hat{U}$  as approximated uncertainty with labelProb.

**return**  $\hat{U}$

---

in each layer  $t$  would satisfy the following inequality:

$$\frac{1}{k^t} \sum_{i=1}^{k^t} (\Pr(y_{o_i}^t | \mathbf{x}; \theta) - \Pr(y_{\ell_i}^t | \mathbf{x}; \theta)) \leq \delta',$$

where  $\delta' \approx 0$  is a small positive constant;  $o_i$  and  $\ell_i$  are the oracle and predicted top- $i$  label.

The proof of Theorem 3.1 is presented in Appendix A. The effectiveness of approximated *total uncertainty* can be also shown in Theorem 3.2

**Theorem 3.2.** *Suppose  $\hat{U}(\mathbf{x}, \ell)$  is the approximated total uncertainty given by Algorithm 1 to estimate label-level total uncertainty for an instance  $\mathbf{x}$  and each label  $\ell$ ;  $\Pr(y_\ell | \mathbf{x}; \theta^{(m)})$  is under a long-tail distribution as shown in Figure 1. With a large enough hyper-parameter  $k$  in beam search, for each individual model, the approximated uncertainty  $\hat{U}(\mathbf{x})$  and  $\hat{U}(\mathbf{x}, \ell)$  would satisfy the following inequality:*

$$|U(\mathbf{x}, \ell) - \hat{U}(\mathbf{x}, \ell)| \leq \delta',$$

where  $\delta' \approx 0$  is a small positive constant.

Note that we show the proof of Theorem 3.2 in Appendix B. Other uncertainty metrics like *knowledge uncertainty* and *probability variance* can also reach similar properties.

### 3.4 Time Complexity Analysis

Suppose  $d$  is the depth of tree-based models in the ensemble; matrix multiplication is implemented with sparse matrices; with  $B$ -ary clustering, the cluster number  $K_t$  is  $\min(B^t, L)$ , where  $B$  is a small constant;  $T_\theta$  is the time to compute  $\Pr(y | \mathbf{x}; \theta)$ . For simplicity, we let  $k = b$ ;  $T_{\theta^{(m)}} = T_\theta$  for all individual models. The time complexity of Algorithm 1 for uncertainty approximation is

$$O\left(MdbT_\theta \max\left(B, \frac{L}{B^{d-1}}\right)\right).$$

If  $B$  and  $d$  are decided such that  $d = O(\log_B L)$  (i.e.,  $\frac{L}{B^{d-1}}$  is a small constant), the overall time complexity would be

$$O(MvbT_\theta \log L).$$

Compared to the complexity of naïve approach to compute the predictions over all labels for each model  $O(M \times L \times T_\theta)$ , our algorithm significantly reduces the computational time from linear to sub-linear to the extremely large size of the label spaces  $L$ .

### 3.5 Generating XMC Model Ensembles

Our framework is general for arbitrary methods of generating model ensembles. In this work, as examples, we adopt two ensemble generation approaches from data and model perspectives without manipulating model weights, including bagging and boosting.

**Bootstrap Aggregating (Bagging).** Bagging is one of the most popular ensemble meta-algorithms to enhance the stability of machine learning models [8]. In other words, individual models in the bagging ensemble can be representative of the diversity of model predictions so that the ensemble model can reduce variance [17, 21].

For each individual model  $\Pr(y | x; \theta^{(m)})$ , we learn the model parameters  $\theta^{(m)}$  with a bootstrapped dataset  $\mathcal{D}^{(m)} \sim \mathcal{D}$  sampled from the training dataset  $\mathcal{D}$  with replacement. Formally, for each label  $\ell$ , prediction of individual models can be presented with notions described in Section 2.2 as:

$$\Pr(y_\ell | x; \theta^{(m)}) = \Pr(y_\ell | x, \mathcal{D}^{(m)}), \mathcal{D}^{(m)} \sim \mathcal{D}.$$

**Boosting with Hard Negatives.** In addition to deriving ensemble models from the data perspective, the model perspective could be also important. As mentioned in Section 2.2, negative example selection plays a critical role in training XMC models. Inspired by the matcher-aware negatives [54], we propose to iteratively boost XMC models by leveraging earlier models to derive hard negatives as new training samples. Precisely, given sets of model parameters  $\{\theta^{(t)}\}_{t < m}$  trained before the iteration  $m$ , the boosted XMC model  $\Pr(y | x; \{\theta^{(t)}\}_{t \leq m})$  for the next iteration can be learned by a linear combination of the previous boosted model and the new individual model trained with hard negatives as:

$$(1 - \alpha) \Pr(y | x; \{\theta^{(t)}\}_{t < m}) + \alpha \Pr(y | x; \theta^{(m)}),$$

where  $0 \leq \alpha \leq 1$ . The new individual model  $\Pr(y | x; \theta^{(m)})$  learns the model parameters from new training data enhanced by hard negatives as:

$$\Pr(y | x, D' \cup \text{Hard}(D, \{\theta^{(t)}\}_{t < m})),$$

where  $D'$  is the default selected negatives (e.g., teacher forcing negatives); the function  $\text{Hard}(D, \{\theta^{(t)}\}_{t < m})$  derives the top  $k_{\text{hard}}$  false positive predictions given by the previous boosted model  $\Pr(y | x; \{\theta^{(t)}\}_{t < m})$ .

**Boosted Bagging.** Although the above two approaches can be independently adopted for generating ensembles, they can be jointly utilized since they rely on different perspectives. Here we denote the combination of *Bagging* and *Boosting* as *Boosted Bagging* in our

**Table 1: The statistics of six experimental datasets. Note that  $n_{\text{train}}$ ,  $n_{\text{test}}$  are the numbers of training and testing instances.  $L$  is the number of labels while  $\bar{L}$  the average number of labels per instance.  $\bar{n}$  the average number of instances per label.**

Dataset	$n_{\text{train}}$	$n_{\text{test}}$	$d$	$L$	$\bar{L}$	$\bar{n}$
Eurlex-4K	15,449	3,865	186,104	3,956	5.30	20.79
Wiki10-31K	14,146	6,616	101,938	30,938	18.64	8.52
Amazon-670K	490,449	153,025	135,909	670,091	5.45	3.99
AmazonCat-13K	1,186,239	306,782	203,882	13,330	5.04	448.57
Wiki-500K	1,779,881	769,421	2,381,304	501,070	4.75	16.86
Amazon-3M	1,717,899	742,507	337,067	2,812,281	36.04	22.02

experiments by deriving boosting models based on bootstrapped data and corresponding hard negatives.

## 4 EXPERIMENTS

In this section, we evaluate our uncertainty quantification framework in three tasks about predictive accuracy and uncertainty quality, as well as its efficiency.

### 4.1 Experimental Settings

**Datasets.** In this paper, we adopt six public benchmark extreme multi-label text classification datasets [52], including Eurlex-4k, Wiki10-31K, Amazon-670K, AmazonCat-13K, Wiki-500K, and Amazon-3M, as shown in Table 1. We use the sparse tfidf representations as features and training/testing data splits, which are consistent with existing studies in this field [10, 25, 52, 55].

**Base XMC Models.** We first consider XR-LINEAR in PECOS [54] with sparse tfidf representations as our base model since XR-LINEAR is one of the most popular tree-based XMC models with high flexibility and scalability to enormous output spaces and applied to various domains like extreme text classification [35] and product search [9]. We then adopt XR-TRANSFORMER [55], one of the state-of-the-art text XMC models, to show the potential of our framework to be applied in deep XMC models.

**Evaluation Tasks and Metrics.** We consider three evaluation tasks: (1) predictive performance, (2) misclassification detection, and (3) out-of-distribution (OOD) detection. Task 1 evaluates prediction accuracy with precision and recall metrics on top-ranked labels. Task 2 assesses label-level uncertainty by detecting incorrectly ranked labels with uncertainty scores. Task 3 appraises instance-level uncertainty by identifying testing instances out of training distributions with uncertainty scores. Tasks 2 and 3 utilize area under the ROC curve (AUROC) [23] as the evaluation metric.

**Baselines.** We mainly compare with single models. For Tasks 2 and 3, we adopt *Energy* [34] and *JointEnergy* [47], the state-of-the-art OOD detection methods for conventional multi-class and multi-label classification, to derive baseline uncertainty scores. Besides, since our approximation framework can be applied to arbitrary model ensembles. We consider Monte-Carlo dropout (*MC Dropout*) [20] with a 5% dropout rate as a comparative baseline to verify the effectiveness of our ensemble generation.

Dataset	P@1	P@3	P@5	R@1	R@3	R@5	P@1	P@3	P@5	R@1	R@3	R@5	P@1	P@3	P@5	R@1	R@3	R@5
	Eurlex-4K						Wiki10-31K						Amazon-670K					
Single Model	81.76	69.02	57.61	16.52	41.00	55.99	84.05	73.06	64.09	4.96	12.73	18.33	44.25	39.34	35.70	9.21	22.82	33.46
MC Dropout	81.91	68.95	57.67	16.55	40.95	56.07	84.04	73.06	64.07	4.96	12.74	18.33	44.29	39.37	35.70	9.21	22.84	33.47
Boosting	<b>82.74</b>	69.55	<b>58.16</b>	<b>16.74</b>	41.33	<b>56.56</b>	84.33	<b>73.67</b>	<b>64.58</b>	4.98	<b>12.85</b>	<b>18.50</b>	44.54	39.51	35.80	9.31	22.96	33.58
Bagging	81.79	69.05	57.63	16.55	41.03	56.02	84.36	73.02	63.98	4.97	12.72	18.30	44.34	39.39	35.78	9.23	22.85	33.54
Boosted Bagging	82.69	<b>69.68</b>	58.14	16.73	<b>41.43</b>	<b>56.56</b>	<b>84.55</b>	73.58	64.53	<b>5.00</b>	12.82	18.47	<b>44.64</b>	<b>39.65</b>	<b>35.96</b>	<b>9.32</b>	<b>23.03</b>	<b>33.72</b>
	AmazonCat-13K						Wiki-500K						Amazon-3M					
Single Model	92.60	78.43	63.79	26.17	59.37	74.66	67.05	48.02	<b>37.52</b>	21.84	39.85	<b>48.08</b>	46.42	43.60	41.52	2.89	7.17	10.59
MC Dropout	92.61	78.44	63.79	26.17	59.38	74.67	67.01	47.97	37.47	21.81	39.81	48.02	46.42	43.58	41.48	2.88	7.16	10.57
Boosting	93.01	78.81	<b>64.08</b>	26.30	59.68	<b>75.02</b>	<b>67.85</b>	<b>48.06</b>	37.33	<b>22.17</b>	<b>39.98</b>	47.96	47.05	<b>44.29</b>	<b>42.16</b>	<b>3.03</b>	<b>7.48</b>	<b>11.01</b>
Bagging	92.77	78.51	63.87	26.24	59.43	74.75	66.92	47.89	37.43	21.77	39.74	47.98	46.64	43.76	41.66	2.90	7.18	10.61
Boosted Bagging	<b>93.09</b>	<b>78.82</b>	<b>64.08</b>	<b>26.34</b>	<b>59.69</b>	<b>75.02</b>	67.69	48.01	37.33	22.08	39.92	47.97	<b>47.11</b>	<b>44.29</b>	42.14	3.02	7.43	10.93

**Table 2: Predictive performance of different methods in percentage (%) over six experimental datasets. P@k and R@k represent precision and recall metrics with top-k predicted labels.**

Dataset	Energy	MC Dropout	Boosting			Bagging			Boosted Bagging		
			PV	TU	KU	PV	TU	KU	PV	TU	KU
Eurlex-4K	83.54	93.54	93.37	93.51	94.34	94.19	93.47	93.73	<b>94.39</b>	93.53	94.18
Wiki10-31K	87.28	86.48	86.44	87.36	87.28	86.30	87.14	87.09	<b>89.46</b>	87.18	87.14
Amazon-670K	80.73	95.94	95.02	95.65	95.55	<b>96.12</b>	95.67	95.67	95.82	95.64	95.49
AmazonCat-13K	85.15	95.50	<b>95.87</b>	95.58	95.47	95.68	95.52	95.53	95.35	95.73	95.70
Wiki-500K	80.10	93.59	92.15	93.42	<b>94.07</b>	93.95	93.52	93.76	93.33	93.43	93.95
Amazon-3M	77.04	77.00	75.46	<b>77.42</b>	75.18	76.03	76.95	76.36	75.87	77.26	76.03

**Table 3: The misclassification detection performance of different methods in average AUROC (%) for evaluating the quality of label-level uncertainty. PV, TU, and KU represent *Probability Variance*, *Total Uncertainty*, and *Knowledge Uncertainty*.**

**Experimental Details.** Experiments described in Sections 4.2, 4.3, 4.4, and 4.5 without a need of GPUs are conducted on an AWS x1.32xlarge instance with 128 CPU cores based on four Intel Xeon E7-8880 v3 2.30GHz processors and 1,952 GiB memory. Experiments described in Section 4.6 with needs of GPUs are conducted on an AWS p3dn.24xlarge instance with 96 CPU cores based on four Intel Xeon Platinum 8175M 2.50GHz processors, 96 GiB memory, and 8 NVIDIA V100 Tensor Core GPUs with 32 GB of memory each.

For XR-Linear as our base XMC model, we establish hierarchical label trees based on Positive Instance Feature Aggregation (PIFA) [54] and 8-means (i.e.,  $B_t = 8$ ). After training, model weights smaller than  $1e-3$  are ignored for reducing disk space consumption, following the settings of XR-LINEAR [54]. For inference, the beam size  $b$  and the number of top-ranked labels  $k$  in beam search are set as 50 and 100. For the *Bagging* ensemble approach, we bootstrap datasets with the identical size to the training dataset for training individual models. For the *Boosting* ensemble approach, we set the hyper-parameter  $\alpha = 0.5$ . For all ensemble-based approaches in the experiments (i.e., all methods except the single model), the number of model ensembles  $M$  is 10 for ensemble generation. For XR-TRANSFORMER [55] as our base XMC model, we follow the instructions in the original paper to train every individual XR-TRANSFORMER model. All deep learning models using transformers use the bert-base-uncased model as the pretrained model [16].

## 4.2 Task 1: Predictive Performance

Table 2 shows the predictive performance on six experimental datasets. Compared to the single model, *MC Dropout* does not improve the predictive accuracy. This could be because of sparse feature representations in XMC so that dropping model weights is less likely to affect inference. In contrast, our ensemble approaches outperform using only a single model in most of the metrics since we consider the uncertainty from both data and model perspectives. For example, *Boosted Bagging* outperforms both *Single Model* and *MC Dropout* by 1.5% and 4.8% in P@1 and R@1 for the Amazon-3M dataset. An interesting observation is: although *Boosted Bagging* performs better for smaller datasets, *Boosting* becomes the best while it comes to larger datasets, such as Wiki-500K and Amazon-3M. The reason could be fewer instances, i.e., less knowledge, in training data for each individual model, so those weak models could be incapable of constructing a strong ensemble.

## 4.3 Task 2: Misclassification Detection

The task of misclassification detection aims to evaluate the quality of estimated label-level uncertainty. Table 3 demonstrates the performance of all methods in average AUROC over six experimental datasets. All ensemble-based models outperform *Energy* using single model using only entropy as the uncertainty indicator, showing the importance of sampling different model parameters in the manner of Bayesian ensembles [19, 36, 38]. Our ensemble approaches still outperform all baseline methods over all datasets in misclassification detection. For instance, PV using *Boosted Bagging* outperforms *Energy* and *MC Dropout* by 2.5% and 3.4%. The

Training Dataset	OOD Dataset	Joint Energy	MC Dropout	Boosting			Bagging			Boosted Bagging		
				PV	TU	KU	PV	TU	KU	PV	TU	KU
Wiki10-31K	Eurlex-4K	96.89	97.11	97.17	96.71	95.76	<b>97.21</b>	96.97	96.76	97.08	96.81	96.58
	Amazon-670K	95.55	<b>96.66</b>	96.53	95.52	95.13	96.58	95.68	95.39	96.56	95.70	95.35
	AmazonCat-13K	94.60	<b>95.91</b>	95.76	94.63	93.99	95.82	94.75	94.45	95.82	94.81	94.37
	Wiki-500K	91.01	90.43	90.70	90.71	89.10	90.91	91.12	<b>91.19</b>	90.92	91.09	91.13
	Amazon-3M	94.89	<b>96.28</b>	96.17	94.87	94.31	96.14	95.06	94.63	96.15	95.10	94.48
Amazon-670K	Eurlex-4K	98.68	98.67	98.79	<b>98.85</b>	98.65	98.75	98.62	98.71	98.83	98.76	98.69
	Wiki10-31K	71.01	72.85	72.83	71.27	70.90	73.22	71.28	71.21	<b>73.31</b>	71.38	71.26
	AmazonCat-13K	48.38	50.62	<b>51.11</b>	48.41	49.50	50.63	48.39	48.34	50.63	48.17	48.84
	Wiki-500K	88.29	88.78	88.55	88.64	88.04	89.27	88.36	88.50	<b>89.58</b>	88.88	88.43
	Amazon-3M	64.27	66.48	<b>67.15</b>	65.08	64.58	66.67	64.31	64.43	66.59	64.49	64.55

**Table 4: The out-of-distribution (OOD) detection performance of different methods in AUROC (%) for evaluating the quality of instance-level uncertainty. PV, TU, and KU represent *Probability Variance*, *Total Uncertainty*, and *Knowledge Uncertainty*.**

Dataset	Approach	Time (Minutes)	Boosted Bagging		
			PV	TU	KU
Misclassification Detection					
Eurlex-4K	Naïve	49.12	97.35	97.17	97.23
	Beam Search	<b>1.62 (30.32x)</b>	94.39	93.53	94.18
Wiki10-31K	Naïve	671.18	91.06	90.77	91.04
	Beam Search	<b>6.08 (110.39x)</b>	89.46	87.18	87.14
OOD Detection (Trained on Wiki10-31K)					
Eurlex-4K	Naïve	1135.82	97.13	97.10	96.91
	Beam Search	<b>18.42 (61.66x)</b>	97.08	96.81	96.58
Amazon-670K	Naïve	2501.06	97.02	96.16	96.28
	Beam Search	<b>62.87 (39.78x)</b>	96.56	95.70	95.35

**Table 5: The execution time with the performance of misclassification detection and OOD detection for the *Boosted Bagging* method based on the naïve approach and beam search approximation for uncertainty quantification. PV, TU, and KU represent *Probability Variance*, *Total Uncertainty*, and *Knowledge Uncertainty*.**

observation on performance among our ensemble approaches over different dataset sizes as described in the task of predictive performance still holds here. We also notice that PV performs better than TU and KU for smaller datasets while TU and KU work better for larger ones. It can be because the predicted probabilities could be more accurate and less likely to require calibration with more training data.

#### 4.4 Task 3: Out-of-Distribution (OOD) Detection

In the task of OOD detection, we evaluate the quality of estimated instance-level uncertainty. Table 4 shows the performance in AUROC for all methods with six datasets. Similar to the experimental results in misclassification detection, ensemble-based methods outperform *JointEnergy* using single models in most cases. Another similar observation is: with the smaller training dataset like Wiki10-31K and smaller model sizes, *MC Dropout* and *Bagging* perform better. On the other hand, *Boosting* and *Boosted Bagging* are the best methods when it comes to using Amazon-670K as the training dataset. Moreover, general performance on distinguishing AmazonCat-13K and Amazon-3M from Amazon-670K is lower

Dataset	P@1	P@3	P@5	R@1	R@3	R@5
	Eurlex-4K					
Single Model	87.17	74.51	61.51	17.71	44.45	59.86
Boosted Bagging	<b>88.10</b>	<b>75.71</b>	<b>62.21</b>	<b>17.92</b>	<b>45.10</b>	<b>60.53</b>
Wiki10-31K						
Single Model	87.89	78.69	69.08	5.24	13.84	19.89
Boosted Bagging	<b>88.54</b>	<b>79.68</b>	<b>70.31</b>	<b>5.29</b>	<b>14.01</b>	<b>20.24</b>
Amazon-670K						
Single Model	49.00	43.68	39.81	10.30	25.49	37.49
Boosted Bagging	<b>49.95</b>	<b>44.58</b>	<b>40.67</b>	<b>10.49</b>	<b>25.99</b>	<b>67.26</b>

**Table 6: Predictive performance with XR-TRANSFORMER as the base XMC model in percentage (%) on three datasets.**

Dataset	Energy	Boosted Bagging		
		PV	TU	KU
Eurlex-4K	90.10	91.74	<b>92.75</b>	92.03
Wiki10-31K	88.19	88.29	<b>89.28</b>	88.69
Amazon-670K	91.30	92.46	<b>95.04</b>	93.10

**Table 7: The misclassification detection performance with XR-TRANSFORMER as the base XMC model on three datasets. PV, TU, and KU represent *Probability Variance*, *Total Uncertainty*, and *Knowledge Uncertainty*.**

because of similar dataset distributions. However, our proposed methods can still outperform baseline methods. We further found that variance-based quantification methods (i.e., *MC Dropout* and *PV*) generally perform better than entropy-based approaches.

#### 4.5 Approximation Efficiency

Table 5 states the execution time with the performance of Tasks 2 and 3 for *Boosted Bagging* based on the naïve approach and our beam search approximation for uncertainty quantification. As a result, beam search approximation is significantly faster than the naïve approach. For example, in misclassification detection, our approximation obtains 110.39x and 30.32x speedups with only 1.7% and 3.0% drop in AUROC using PV on Wiki10-31K and Eurlex-4K datasets. In OOD detection, after training XMC models with the Wiki10-31K dataset, beam search is 61.66x and 39.78x faster than

Method	P@1	P@3	P@5	P@1	P@3	P@5	P@1	P@3	P@5
	Eurlex-4K			Wiki10-31K			Amazon-670K		
AnnexML [44]	79.66	64.94	53.52	86.46	74.28	64.20	42.09	36.61	32.75
DiSMEC [3]	83.21	70.39	58.73	84.13	74.72	65.94	44.78	39.72	36.17
PfasteXML [24]	73.14	60.16	50.54	83.57	68.61	59.10	36.84	34.23	32.09
Parabel [39]	82.12	68.91	57.89	84.19	72.46	63.37	44.91	39.77	35.98
eXtremeText [50]	79.17	66.80	56.09	83.66	73.28	64.51	42.54	37.93	34.63
Bonsai [26]	82.30	69.55	58.35	84.52	73.76	64.69	45.58	40.39	36.60
XML-CNN [32]	75.32	60.14	49.21	81.41	66.23	56.11	33.41	30.00	27.42
AttentionXML [52]	85.49	73.08	61.10	87.05	77.78	68.78	45.66	40.67	36.94
X-Transformer [10]	85.82	73.23	61.18	87.79	78.43	69.74	46.45	40.82	36.71
LightXML [25]	85.60	74.10	62.00	87.84	77.26	67.99	47.30	42.20	38.50
XR-TRANSFORMER [55]	87.17	74.51	61.51	87.89	78.69	69.08	49.00	43.68	39.81
XR-TRANSFORMER with <i>Boosted Bagging</i>	<b>88.10</b>	<b>75.71</b>	<b>62.21</b>	<b>88.54</b>	<b>79.68</b>	<b>70.31</b>	<b>49.95</b>	<b>44.58</b>	<b>40.67</b>

**Table 8: Predictive performance of various XMC models and our approach using XR-TRANSFORMER as the base XMC model on Eurlex-4K and Wiki10-31K. P@k represents precision with top-k predicted labels.**

the naïve method with only 0.1% and 0.5% performance loss in AUROC using PV to identify Eurlex-4K and Amazon-670K as OOD datasets.

#### 4.6 Performance on Deep XMC Models

To further demonstrate the potential of our proposed approach with deep learning, we adopt XR-TRANSFORMER, which is the state-of-the-art XMC model based text data and transformers [16, 45], as the base XMC model. Table 6 shows the predictive performance with XR-TRANSFORMER as the base XMC model on three datasets. Our ensemble-based method using *Boosted Bagging* outperforms the single XR-TRANSFORMER model across all metrics. For evaluating the quality of estimated uncertainty, Table 7 provides the performance on misclassification detection. All uncertainty metrics based on *Boosted Bagging* are better than *Energy* using a single model. Besides, XR-TRANSFORMER with uncertainty quantification using our approach can actually further yield better state-of-the-art results. Table 8 shows the predictive performance of various XMC models and our approach *Boosted Bagging* with XR-TRANSFORMER as the base XMC model, where comparative methods do not consider uncertainty quantification. As a result, after applying our ensemble-based uncertainty quantification approach to XR-TRANSFORMER, we can obtain state-of-the-art performance in most of the metrics. This further demonstrate the benefits of modeling uncertainty in the XMC task.

## 5 RELATED WORK

**Ensemble-based Uncertainty Quantification.** Uncertainty quantification aims at characterizing (and reducing) uncertainties in computational applications [46]. Instead of using a single model capturing only *data uncertainty* (or aleatoric uncertainty) [18, 19, 36], ensemble-based approaches generate ensembles of models so that the overall estimated uncertainty can be decomposed into *data uncertainty* and *knowledge uncertainty* (or epistemic uncertainty) [15]. For example, Gal and Ghahramani [20] collect model ensembles by dropping model weights of neural networks during training; Malinin et al. [38] generate ensembles of gradient boosting decision

trees with intermediate individual models; Malinin and Gales [37] apply the entropy chain rule and sampling to collect an ensemble for autoregressive structured prediction; Deep Ensemble [29] derives the model ensemble by varying initial weights of neural models. These ensemble-based uncertainty quantification methods have been applied to real-world applications in various domains, such as medicine diagnosis [13, 40] and computer vision [33, 56]. In addition, ensemble models are also usually capable of improving predictive performance. However, although ensemble-based approaches have obtained many successes in different fields, none of the previous studies focuses on the XMC task due to the challenges introduced in Section 1.

**Partition-based Extreme Multi-label Classification.** Computational challenge is always the hurdle of the XMC problem. Even with sparse linear XMC models [4] comparatively more lightweight, naïve one-versus-rest (OVR) methods [51] with a linear inference time could be still too slow to be applied in real-world applications. Partition-based methods are one of the most popular approaches to addressing the efficiency and scalability of XMC models [9]. By introducing different partitioning techniques, the enormous label spaces can be reduced into hierarchical label trees so that the exhausting process on examining all labels can be replaced by efficient semantic indexing as mentioned in Section 2.2. Parabel [39] first establishes balanced 2-means label trees with instance-induced label features. Accordingly, various successors propose diverse partitioning and indexing methods for improvements, such as eXtremeText [50], Bonsai [26], and PECOS [54]. Moreover, with longer training and inference time, partition-based methods can obtain state-of-the-art accuracy by utilizing deep neural encoders, such as AttentionXML [52], LightXML [25], and XR-TRANSFORMER [55], for time-insensitive applications. In particular, XR-LINEAR in PECOS and XR-TRANSFORMER are the state-of-the-art sparse linear and deep XMC models, and treated as the base XMC models in our experiments. For simplicity, partition-based methods are called *tree-based methods* in this work.

## 6 CONCLUSIONS

In this paper, we are the pioneer of studying uncertainty quantification for eXtreme Multi-label Classification (XMC). We first propose to generate ensembles with the techniques of bootstrapping and boosting for estimating uncertainty from different perspectives.. Besides, we also come up with two different uncertainty levels, including *label-level* and *instance-level* uncertainty. To overcome the computational issues over the enormous label spaces, we further suggest to approximating uncertainty with beam search under long-tail probability distribution in XMC problems. In experiments, we demonstrate our proposed approaches not only obtain superior predictive performance, but also result in high-quality uncertainty estimation in both label-level and instance-level uncertainty, compared to baseline methods. Moreover, our framework can deliver better state-of-the-art XMC results after considering uncertainty quantification and using deep XMC models as base models.

## APPENDIX

In the Appendix, we prove the theorems in this paper.

### A PROOF OF THEOREM 3.1

PROOF. Without loss of generality, we assume a pair of  $(o_i, \ell_i)$  resulting a positive regret, where  $o_i$  is out of beam search results;  $P(y_{o_i}^t | \mathbf{x}; \boldsymbol{\theta}) > P(y_{\ell_i}^t | \mathbf{x}; \boldsymbol{\theta})$ . Based on Equation (4), we have:

$$P(y_{o_i}^t | \mathbf{x}; \boldsymbol{\theta}) = P(y_{o_i}^t | \mathbf{x}, c_{o_i}^t; \boldsymbol{\theta}) \times P(c_{o_i}^t | \mathbf{x}; \boldsymbol{\theta}),$$

where  $c_{o_i}^t$  is the corresponding cluster in the previous layer. Since  $o_i$  is out of beam search results,  $o_i$  is ranked after the  $k_{t-1}$ -th position among  $P(y^{t-1} | \mathbf{x}; \boldsymbol{\theta})$ . As mentioned in Section 2.2,  $P(c^t | \mathbf{x}; \boldsymbol{\theta})$  represents  $P(y^{t-1} | \mathbf{x}; \boldsymbol{\theta})$  in the recursive manner of tree-based XMC models, so  $P(c^t | \mathbf{x}; \boldsymbol{\theta})$  is also under a long-tail distribution. Hence, if the beam size  $k_{t-1}$  is large enough, we would have:

$$P(c_{o_i}^t | \mathbf{x}; \boldsymbol{\theta}) \leq \delta',$$

where  $\delta' \approx 0$  is a small positive constant. Moreover, the probability  $P(y_{o_i}^t | \mathbf{x}; \boldsymbol{\theta})$  is also bounded as:

$$P(y_{o_i}^t | \mathbf{x}; \boldsymbol{\theta}) \leq P(y_{o_i}^t | \mathbf{x}, c_{o_i}^t; \boldsymbol{\theta}) \times \delta' \leq \delta'.$$

Based on the assumption, we then have:

$$P(y_{o_i}^t | \mathbf{x}; \boldsymbol{\theta}) - P(y_{\ell_i}^t | \mathbf{x}; \boldsymbol{\theta}) \leq \delta'. \quad (5)$$

By extending Equation (5) to all positions, we would have:

$$\frac{1}{k^t} \sum_{i=1}^{k^t} (P(y_{o_i}^t | \mathbf{x}; \boldsymbol{\theta}) - P(y_{\ell_i}^t | \mathbf{x}; \boldsymbol{\theta})) \leq \frac{1}{k^t} \sum_{i=1}^{k^t} \delta' = \delta'. \quad \square$$

### B PROOF OF THEOREM 3.2

For simplicity, in our proofs,  $\hat{P}(y_i | \mathbf{x}, \boldsymbol{\theta}^{(m)})$  represents the  $i$ -th top probability approximated by beam search in Algorithm 1 as:

$$\hat{P}(y_i | \mathbf{x}, \boldsymbol{\theta}^{(m)}) = \begin{cases} P(y_\ell | \mathbf{x}; \boldsymbol{\theta}^{(m)}) & , \text{ if } i \leq k, \\ \delta & , \text{ else} \end{cases}, \text{ where } \delta \approx 0.$$

Here we start from proving the following Lemma B.1.

**Lemma B.1.** *For each set of model parameters  $\boldsymbol{\theta}^{(m)}$ , suppose  $P(y_i | \mathbf{x}; \boldsymbol{\theta}^{(m)})$  denotes the  $i$ -th greatest probability in  $\{P(y_\ell | \mathbf{x}; \boldsymbol{\theta}^{(m)}) |$*

$\ell \in \mathcal{L}\}$ . If probabilities  $P(y_\ell | \mathbf{x}; \boldsymbol{\theta}^{(m)})$  is under a long-tail distribution as shown in Figure 1, with a large enough number  $k$ , for any label  $\ell$  we have:

$$\left| (-\ln P(y_\ell | \mathbf{x}; \boldsymbol{\theta}^{(m)})) - (-\ln \hat{P}(y_\ell | \mathbf{x}; \boldsymbol{\theta}^{(m)})) \right| \leq \delta',$$

, where  $\delta' \approx 0$  is a small positive constant;

PROOF. Since  $P(y_\ell | \mathbf{x}; \boldsymbol{\theta}^{(m)})$  is under a long-tail distribution as shown in Figure 1, with a large enough  $k$ , we have:

$$P(y_i | \mathbf{x}; \boldsymbol{\theta}^{(m)}) \approx 0 \approx \delta, \forall i > k.$$

For the top cases, where  $i \leq k$ , it is trivial that for any  $\delta' > 0$ , we have:

$$\left| (-\ln P(y_\ell | \mathbf{x}; \boldsymbol{\theta}^{(m)})) - (-\ln \hat{P}(y_\ell | \mathbf{x}; \boldsymbol{\theta}^{(m)})) \right| = 0 \leq \delta'.$$

For the other cases, i.e.,  $i > k$ , we have:

$$\begin{aligned} & \left| (-\ln P(y_\ell | \mathbf{x}; \boldsymbol{\theta}^{(m)})) - (-\ln \hat{P}(y_\ell | \mathbf{x}; \boldsymbol{\theta}^{(m)})) \right| \\ &= \left| \ln \hat{P}(y_\ell | \mathbf{x}; \boldsymbol{\theta}^{(m)}) - \ln P(y_\ell | \mathbf{x}; \boldsymbol{\theta}^{(m)}) \right| \\ &= \left| \ln \frac{\hat{P}(y_\ell | \mathbf{x}; \boldsymbol{\theta}^{(m)})}{P(y_\ell | \mathbf{x}; \boldsymbol{\theta}^{(m)})} \right| \\ &= \left| \ln \frac{\delta}{P(y_\ell | \mathbf{x}; \boldsymbol{\theta}^{(m)})} \right| \\ &\approx \left| \ln \frac{\delta}{\delta} \right| = |\ln 1| = 0. \end{aligned}$$

Therefore, it must exist a small positive constant  $\delta'$  so that

$$\left| (-\ln P(y_\ell | \mathbf{x}; \boldsymbol{\theta}^{(m)})) - (-\ln \hat{P}(y_\ell | \mathbf{x}; \boldsymbol{\theta}^{(m)})) \right| \leq \delta'. \quad \square$$

Based on Lemma B.1, we can prove Theorem 3.2 accordingly.

PROOF.

$$\begin{aligned} |U(\mathbf{x}, \ell) - \hat{U}(\mathbf{x}, \ell)| &= \left| \mathcal{H}[P(y_\ell | \mathbf{x}, \mathcal{D})] - \mathcal{H}[\hat{P}(y_\ell | \mathbf{x}, \mathcal{D})] \right| \\ &= \left| \mathbb{E}_{p(y_\ell | \mathbf{x}, \mathcal{D})} [-\ln P(y_\ell | \mathbf{x}, \mathcal{D})] - \mathbb{E}_{p(y_\ell | \mathbf{x}, \mathcal{D})} [-\ln \hat{P}(y_\ell | \mathbf{x}, \mathcal{D})] \right| \\ &= \left| \mathbb{E}_{p(y_\ell | \mathbf{x}, \mathcal{D})} [(-\ln P(y_\ell | \mathbf{x}, \mathcal{D})) - (-\ln \hat{P}(y_\ell | \mathbf{x}, \mathcal{D}))] \right| \\ &\approx \left| \frac{1}{M} \sum_{m=1}^M \left( (-\ln P(y_\ell | \mathbf{x}; \boldsymbol{\theta}^{(m)})) - (-\ln \hat{P}(y_\ell | \mathbf{x}; \boldsymbol{\theta}^{(m)})) \right) \right| \end{aligned}$$

(By ensemble-based approximation)

$$\leq \frac{1}{M} \sum_{m=1}^M \left| \left( (-\ln P(y_\ell | \mathbf{x}; \boldsymbol{\theta}^{(m)})) - (-\ln \hat{P}(y_\ell | \mathbf{x}; \boldsymbol{\theta}^{(m)})) \right) \right|$$

(By the Minkowski inequality)

$$\leq \frac{1}{M} \sum_{m=1}^M \delta' = \delta' \text{ (By Lemma B.1)} \quad \square$$

## REFERENCES

- [1] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. 2021. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion* (2021).
- [2] Mario Almagro, Raquel Martínez Unanue, Victor Fresno, and Soto Montalvo. 2020. ICD-10 coding of Spanish electronic discharge summaries: an extreme classification problem. *IEEE Access* 8 (2020), 100073–100083.
- [3] Rohit Babbar and Bernhard Schölkopf. 2017. DiSMEC: Distributed sparse machines for extreme multi-label classification. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. 721–729.
- [4] Rohit Babbar and Bernhard Schölkopf. 2019. Data scarcity, robustness and extreme multi-label classification. *Machine Learning* 108, 8 (2019), 1329–1351.
- [5] Tal Baumel, Jumana Nassour-Kassis, Raphael Cohen, Michael Elhadad, and Noémie Elhadad. 2018. Multi-label classification of patient notes: case study on ICD code assignment. In *Workshops at the thirty-second AAAI conference on artificial intelligence*.
- [6] José M Bernardo and Adrian FM Smith. 2009. *Bayesian theory*. Vol. 405. John Wiley & Sons.
- [7] Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, and Prateek Jain. 2015. Sparse Local Embeddings for Extreme Multi-label Classification.. In *NIPS*, Vol. 29. 730–738.
- [8] Leo Breiman. 1996. Bagging predictors. *Machine learning* 24, 2 (1996), 123–140.
- [9] Wei-Cheng Chang, Daniel Jiang, Hsiang-Fu Yu, Choon Hui Teo, Jiong Zhang, Kai Zhong, Kedarnath Kolluri, Qie Hu, Nikhil Shandilya, Vyacheslav Ievgrafov, et al. 2021. Extreme multi-label learning for semantic matching in product search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2643–2651.
- [10] Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit S Dhillon. 2020. Taming Pretrained Transformers for Extreme Multi-label Text Classification. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3163–3171.
- [11] Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. 2020. Posterior Network: Uncertainty Estimation without OOD Samples via Density-Based Pseudo-Counts. *Advances in Neural Information Processing Systems* 33 (2020), 1356–1367.
- [12] Hugh A Chipman, Edward I George, and Robert E McCulloch. 2007. Bayesian ensemble learning. *Advances in neural information processing systems* 19 (2007), 265.
- [13] Lavsen Dahal, Aayush Kafle, and Bishesh Khanal. 2020. Uncertainty Estimation in Deep 2D Echocardiography Segmentation. *arXiv preprint arXiv:2005.09349* (2020).
- [14] Emily Denton, Jason Weston, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. 2015. User conditional hashtag prediction for images. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 1731–1740.
- [15] Stefan Depeweg, Jose-Miguel Hernandez-Lobato, Finale Doshi-Velez, and Steffen Udluft. 2018. Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning. In *International Conference on Machine Learning*. PMLR, 1184–1193.
- [16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 4171–4186.
- [17] Pedro M Domingos. 1997. Why Does Bagging Work? A Bayesian Account and its Implications.. In *KDD*. Citeseer, 155–158.
- [18] Tony Duan, Avati Anand, Daisy Yi Ding, Khanh K Thai, Sanjay Basu, Andrew Ng, and Alejandro Schuler. 2020. NGBoost: Natural gradient boosting for probabilistic prediction. In *International Conference on Machine Learning*. PMLR, 2690–2700.
- [19] Yarin Gal. 2016. *Uncertainty in Deep Learning*. Ph.D. Dissertation. University of Cambridge.
- [20] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*. PMLR, 1050–1059.
- [21] Yasser Ganjisaffar, Rich Caruana, and Cristina Videira Lopes. 2011. Bagging gradient-boosted trees for high precision, low variance ranking models. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. 85–94.
- [22] Zoubin Ghahramani. 2015. Probabilistic machine learning and artificial intelligence. *Nature* 521, 7553 (2015), 452–459.
- [23] Dan Hendrycks and Kevin Gimpel. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136* (2016).
- [24] Himanshu Jain, Yashoteja Prabhu, and Manik Varma. 2016. Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 935–944.
- [25] Ting Jiang, Deqing Wang, Leilei Sun, Huayi Yang, Zhengyang Zhao, and Fuzhen Zhuang. 2021. LightXML: Transformer with Dynamic Negative Sampling for High-Performance Extreme Multi-label Text Classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 7987–7994.
- [26] Sujay Khandagale, Han Xiao, and Rohit Babbar. 2020. Bonsai: diverse and shallow trees for extreme multi-label classification. *Machine Learning* 109, 11 (2020), 2099–2119.
- [27] Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. 2019. BatchBALD: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems* 32 (2019), 7026–7037.
- [28] Abhishek Kumar, Shankar Vembu, Aditya Krishna Menon, and Charles Elkan. 2013. Beam search algorithms for multilabel learning. *Machine learning* 92, 1 (2013), 65–89.
- [29] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. *Advances in Neural Information Processing Systems* 30 (2017).
- [30] Alex M Lamb, Anirudh Goyal Alias Parth Goyal, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. 2016. Professor forcing: A new algorithm for training recurrent networks. In *NeurIPS*. 4601–4609.
- [31] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. 2006. A tutorial on energy-based learning. *Predicting structured data* 1, 0 (2006).
- [32] Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*. 115–124.
- [33] Jeremiah Zhe Liu. 2019. Variable selection with rigorous uncertainty quantification using bayesian deep neural networks. In *Bayesian Deep Learning Workshop at NeurIPS*.
- [34] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. 2020. Energy-based Out-of-distribution Detection. *Advances in Neural Information Processing Systems* 33 (2020).
- [35] Xuanqing Liu, Wei-Cheng Chang, Hsiang-Fu Yu, Cho-Jui Hsieh, and Inderjit Dhillon. 2021. Label disentanglement in partition-based extreme multilabel classification. *Advances in Neural Information Processing Systems* 34 (2021).
- [36] Andrey Malinin. 2019. *Uncertainty estimation in deep learning with application to spoken language assessment*. Ph. D. Dissertation. University of Cambridge.
- [37] Andrey Malinin and Mark Gales. 2020. Uncertainty Estimation in Autoregressive Structured Prediction. In *International Conference on Learning Representations*.
- [38] Andrey Malinin, Liudmila Prokhorenkova, and Aleksei Ustimenko. 2020. Uncertainty in Gradient Boosting via Ensembles. In *International Conference on Learning Representations*.
- [39] Yashoteja Prabhu, Anil Kag, Shrutendra Harsola, Rahul Agrawal, and Manik Varma. 2018. Parabel: Partitioned label trees for extreme classification with application to dynamic search advertising. In *Proceedings of the 2018 World Wide Web Conference*. 993–1002.
- [40] Abhijit Guha Roy, Sailesh Conjeti, Nassir Navab, Christian Wachinger, Alzheimer’s Disease Neuroimaging Initiative, et al. 2019. Bayesian QuickNAT: Model uncertainty in deep whole-brain segmentation for structure-wise quality control. *NeuroImage* 195 (2019), 11–22.
- [41] Wang Ruo-Peng and Xu Hong-Min. 2009. A smoothing function for 1-norm support vector machines. In *2009 Fifth International Conference on Natural Computation*, Vol. 1. IEEE, 450–454.
- [42] Mohammad Hossein Shaker and Eyke Hüllermeier. 2020. Aleatoric and epistemic uncertainty with random forests. *arXiv preprint arXiv:2001.00893* (2020).
- [43] Artem Shelmanov, Evgenii Tsymbalov, Dmitri Puzyrev, Kirill Fedyanin, Alexander Panchenko, and Maxim Panov. 2021. How Certain is Your Transformer?. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. 1833–1840.
- [44] Yukihiko Tagami. 2017. AnnexML: Approximate nearest neighbor search for extreme multi-label classification. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 455–464.
- [45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.
- [46] Warren E Walker, Poul Harremoës, Jan Rotmans, Jeroen P Van Der Sluijs, Marjolein BA Van Asselt, Peter Janssen, and Martin P Krayen von Krauss. 2003. Defining uncertainty: a conceptual basis for uncertainty management in model-based decision support. *Integrated assessment* 4, 1 (2003), 5–17.
- [47] Haoran Wang, Weitang Liu, Alex Bocchieri, and Yixuan Li. 2021. Can multi-label classification networks know what they don’t know? *Advances in Neural Information Processing Systems* 34 (2021).
- [48] Alfred Wehrl. 1978. General properties of entropy. *Reviews of Modern Physics* 50, 2 (1978), 221.
- [49] Shunyao Wu, Yuzhu Chen, Zhiruo Li, Jian Li, Fengyang Zhao, and Xiaoquan Su. 2021. Towards multi-label classification: Next step of machine learning for microbiome research. *Computational and Structural Biotechnology Journal* (2021).

- [50] Marek Wydmuch, Kalina Jasinska, Mikhail Kuznetsov, Róbert Busa-Fekete, and Krzysztof Dembczyński. 2018. A no-regret generalization of hierarchical softmax to extreme multi-label classification. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 6358–6368.
- [51] Ian En-Hsu Yen, Xiangru Huang, Pradeep Ravikumar, Kai Zhong, and Inderjit Dhillon. 2016. PD-Sparse: A primal and dual sparse approach to extreme multi-class and multilabel classification. In *International conference on machine learning*. PMLR, 3069–3077.
- [52] Ronghui You, Zihan Zhang, Ziyi Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2019. AttentionXML: Label Tree-based Attention-Aware Model for High-Performance Extreme Multi-Label Text Classification. *Advances in Neural Information Processing Systems* 32 (2019), 5820–5830.
- [53] Hsiang-Fu Yu, Prateek Jain, Purushottam Kar, and Inderjit Dhillon. 2014. Large-scale multi-label learning with missing labels. In *International conference on machine learning*. PMLR, 593–601.
- [54] Hsiang-Fu Yu, Kai Zhong, Jiong Zhang, Wei-Cheng Chang, and Inderjit S Dhillon. 2020. PECOS: Prediction for enormous and correlated output spaces. *arXiv preprint arXiv:2010.05878* (2020).
- [55] Jiong Zhang, Wei-Cheng Chang, Hsiang-Fu Yu, and Inderjit S Dhillon. 2021. Fast Multi-Resolution Transformer Fine-tuning for Extreme Multi-label Text Classification. In *Advances in Neural Information Processing Systems*.
- [56] Jize Zhang, Bhavya Kaikhura, and T Yong-Jin Han. 2020. Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *International Conference on Machine Learning*. PMLR, 11117–11128.
- [57] Wenbin Zheng, Xiaping Fu, and Yibin Ying. 2014. Spectroscopy-based food classification with extreme learning machine. *Chemometrics and Intelligent Laboratory Systems* 139 (2014), 42–47.