CLUR: Uncertainty Estimation for Few-Shot Text Classification with Contrastive Learning

Jianfeng He
jianfenghe@vt.edu
Virginia Tech
Falls Church, VA, USA

Xuchao Zhang
xuchaozhang@microsoft.com
Microsoft
Redmond, WA, USA

Shuo Lei
slei@vt.edu
Virginia Tech
Falls Church, VA, USA

Abdulaziz Alhamadani
hamdani@vt.edu
Virginia Tech
Falls Church, VA, USA

Fanglan Chen
fanglanc@vt.edu
Virginia Tech
Falls Church, VA, USA

Bei Xiao
bxiao@american.edu
American University
Washington, D.C., USA

Chang-Tien Lu
ctlu@vt.edu
Virginia Tech
Falls Church, VA, USA

ABSTRACT

Few-shot text classification has extensive application where the sample collection is expensive or complicated. When the penalty for classification errors is high, such as early threat event detection with scarce data, we expect to know “whether we should trust the classification results or reexamine them.” This paper investigates the Uncertainty Estimation for Few-shot Text Classification (UEFTC), an unexplored research area. Given limited samples, a UEFTC model predicts an uncertainty score for a classification result, which is the likelihood that the classification result is false. However, many traditional uncertainty estimation models in text classification are unsuitable for implementing a UEFTC model. These models require numerous training samples, whereas the few-shot setting in UEFTC only provides a few or just one support sample for each class in an episode. We propose Contrastive Learning from Uncertainty Relations (CLUR) to address UEFTC. CLUR can be trained with only one support sample for each class with the help of pseudo uncertainty scores. Unlike previous works that manually set the pseudo uncertainty scores, CLUR self-adaptively learns them using our proposed uncertainty relations. Specifically, we explore four model structures in CLUR to investigate the performance of three common-used contrastive learning components in UEFTC and find that two of the components are effective. Experiment results prove that CLUR outperforms six baselines on four datasets, including an improvement of 4.52% AUPR on an RCV1 dataset in a 5-way 1-shot setting. Our code and data split for UEFTC are in https://github.com/he159ok/CLUR_UncertaintyEst_FewShot_TextCls.

CCS CONCEPTS

- **Applied computing → Document analysis**.

KDD ’23, August 6–10, 2023, Long Beach, CA, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0103-0/23/08.
https://doi.org/10.1145/3580305.3599276

KEYWORDS

Uncertainty estimation, few-shot, pseudo labels, contrastive learning

ACM Reference Format:
Jianfeng He, Xuchao Zhang, Shuo Lei, Abdulaziz Alhamadani, Fanglan Chen, Bei Xiao, and Chang-Tien Lu. 2023. CLUR: Uncertainty Estimation for Few-Shot Text Classification with Contrastive Learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD ’23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3580305.3599276

1 INTRODUCTION

Few-shot text classification learns a classifier using limited training texts [2, 48]. Few-shot scenarios often involve a crucial decision on whether or not to trust a model’s results. For example, a state-of-the-art (SOTA) model diagnosing a new disease demands high accuracy but initially has access only to a few descriptions of the condition. One approach to achieve a higher classification accuracy is to recheck the most uncertain results by the experts [25, 68]. Experts are expensive and scarce. Therefore, uncertainty estimation is pivotal in optimizing decision-making and saving expert resources in many few-shot applications. Here, we improve the accuracy of Uncertainty Estimation for Few-shot Text Classification (UEFTC). Specifically, UEFTC quantifies the likelihood of misclassification in scenarios with few samples [1]. UEFTC models should yield high uncertainty scores for misclassified predictions and low uncertainty scores for correct predictions.

However, the few-shot setting in UEFTC makes many uncertainty estimation methods difficult to use. Concretely, compared to traditional uncertainty estimation tasks, the few-shot setting in UEFTC provides only a few support samples or even one support sample (1-shot) per class in each episode [2]. Below, we describe how the current methods in uncertainty estimation cannot tackle UEFTC given the few-shot limitation and how our approach improves it.

[1] We detail the UEFTC task setting in Sec. 3.1
[2] The term "support" is explained in Sec. 3.1

The current uncertainty estimation methods are mainly of three kinds. First, Bayesian Neural Networks (BNN)-based methods learn a distribution over the model parameters [44, 49], or learn a distribution for each semantic class [4, 5, 55]. Due to the few-support-sample limitation in UEFTC, it is difficult to learn a distribution by a few or just one support sample per class. As a result, BNN-based methods are not suitable for addressing UEFTC. The second method is ensemble-based, which trains an uncertainty estimation model with augmentations of data (i.e., Gaussian noise [31], adversarial augmentation [52]) or structures (i.e., depth-based ensemble [1] and structure search [66]). The third method is pseudo-label-based, which uses the pseudo uncertainty scores as ground truth to learn an uncertainty model [25]. Since ensemble-based and pseudo-label-based methods do not require numerous support samples, we adopt these two methods to tackle UEFTC.

Though pseudo-label-based methods have growing application in recent years [6, 60], their current usage in uncertainty estimation [25] has a drawback of manually setting pseudo uncertainty scores as the ground truth. Concretely, He et al. [25] proposes MSD1 for uncertainty estimation of text classification, which manually sets coefficients of mix-up [67] as the pseudo uncertainty scores for an uncertainty model training (explained in Sec. A.1.1). However, the manual-set pseudo uncertainty scores can be inaccurate because we do not know a training sample's ground-truth uncertainty score given a model structure. Due to the few-shot setting, the inaccurate pseudo uncertainty scores impact UEFTC more than tasks using numerous samples because each sample weighs more in UEFTC. In a one-shot setting, an inaccurate pseudo uncertainty score means the unique support sample uses an inaccurate ground truth of uncertainty scores, leading to obvious training bias.

We propose Contrastive Learning from Uncertainty Relations (CLUR) to improve the accuracy of pseudo uncertainty scores. Instead of manually setting pseudo uncertainty scores, CLUR self-adaptively learns them by our proposed uncertainty relations. The uncertainty relations are either equal or unequal relations (i.e., >, <) between the uncertainty of a pair of augmented samples. The uncertainty relations are obtained from data augmentation. Since the usage of data augmentation is continuously optimized in contrastive learning, we design CLUR based on contrastive learning to better use the data augmentation. This is the first time that contrastive learning has been applied in UEFTC. Therefore, we also investigate whether the three commonly used model structures (detach, predictor, and intersection comparison that are introduced in Sec. 3.3) in contrastive learning are effective in UEFTC. Finally, we show that CLUR exceeds six baselines on four datasets. Our contributions are summarized below.

Improving uncertainty estimation by a few or just one support sample per class. To our knowledge, we are the first to solve UEFTC under its few-support-sample limitation. Our proposed CLUR can be trained with one support sample per class in each episode because it takes advantage of ensemble and pseudo-label-based methods. Our solution in UEFTC can also motivate uncertainty estimation in other few-shot applications.

Proposing and using uncertainty relations to self-adaptively learn pseudo uncertainty scores as the ground truth uncertainty. To address the issue of manually setting pseudo uncertainty scores, we generate augmented sample pairs to self-adaptively learn their pseudo uncertainty scores by our proposed uncertainty relations. Unlike current contrastive learning models that only have equal relations (i.e., having the same (=) or different (≠) classes) between the augmented samples, our uncertainty relations include additional unequal relations, that are larger (>) or smaller (<) uncertainty relations among the augmented sample uncertainty.

Investigating the performance of the three common-used contrastive learning components in UEFTC. As the first study to apply contrastive learning in UEFTC, we also design four model structures in CLUR to investigate the performance of three common-used contrastive learning components in UEFTC. We find that two of them are effective in UEFTC, enabling us to optimize CLUR. Future UEFTC models can benefit from our findings.

Conducting extensive experiments and benchmarking the UEFTC. We demonstrate that CLUR effectively outperforms six baselines on four datasets (20News, RCV1, Amazon, and HuffPost), including an improvement of 4.52% AUPR on an RCV1 dataset in a 5-way 1-shot setting. We release our code as UEFTC benchmark.

2 RELATED WORK

General methods for uncertainty estimation. There are mainly three uncertainty estimation methods: Bayesian Neural Network (BNN)-based [36], deep-ensemble-based [66], and pseudo-label-based [25]. BNN is a neural network with a prior distribution on model weights or dataset category distributions. As an approximation of BNN, Monte Carlo dropout [14, 34] uses dropout in the model in an ensemble way. Building on BNN, a recent study uses inducing matrices to assist in approximating posterior inference [53]. BNN can handle node classification as well [55]. Based on BNN, Evidential Neural Networks (ENN) [30, 39] calculate Dirichlet distributions, which also need numerous training samples to learn accurate distributions. The deep-ensemble-based method trains a fixed architecture with augmentations of data [31, 52] or structures [1, 66]. In addition, Gales and Malinin [15] scale seq2seq tasks by BNN and deep ensembles. As for the pseudo-label-based method [25], it generates the pseudo uncertainty scores for a training sample given a model. Since BNN-based methods usually require numerous samples, our CLUR is a combination of the deep-ensemble-based and the pseudo-label-based methods, where we use contrastive learning to connect these two methods.

Uncertainty estimation for text classification. It focuses either on the training or the testing data. For example, Wang et al. [59] annotate unlabeled samples with higher uncertainty for training. For testing, it mainly has two tasks: *OOD detection* [13, 24, 38] for predictions, such as Hu and Khan [29]; and *misclassified result detection*, where testing samples are in-domain, such as Zhang et al. [68] and He et al. [25]. Zhang et al. [68] use dropout sampling for uncertainty scores. Three modules are proposed in He et al. [25], where MSD3 calculates the sample distributions and is not applicable to UTFTC. Compared to them, UEFTC addresses misclassified result detection. Different from models requiring many training samples [25, 68], CLUR is trainable with one support sample per class in each episode. To our knowledge, we are the first to estimate uncertainty for few-shot text misclassification detection.

Few-shot text classification. Few-shot text classification has received increasing attention in recent years [40]. The few-shot text

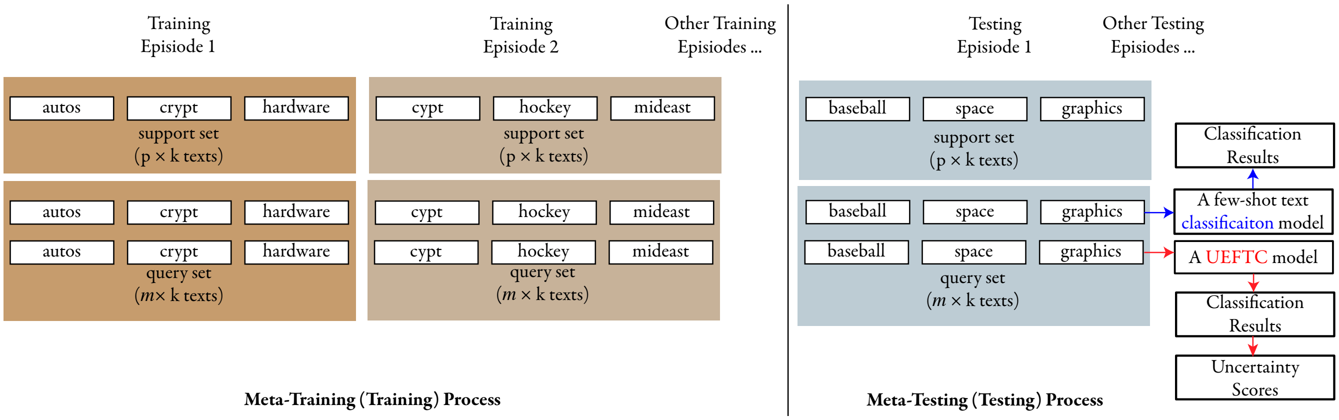


Figure 1: Diagram of UEFTC in sample splits, where $k = 3$, $p = 1$, and $m = 2$. Each tag in the diagram represents a text sample with a respective class. During the meta-training (training) process, a UEFTC model learns via the loss over the query samples in each training episode. The loss functions expect both accurate classification results and uncertainty scores. During the meta-testing (testing) process, the UEFTC model predicts a classification result and an uncertainty score for each query sample from each testing episode. We evaluate UEFTC by the performance of uncertainty scores of query sample classification results from each testing episode. The episodes drawn in this diagram are also applied to few-shot text classification. Compared to few-shot text classification, a UEFTC model additionally targets accurate uncertainty scores besides classification results.

classification models are mainly in two categories: transfer-learning-based and meta-learning-based methods. The transfer-learning-based methods transfer well-learned knowledge to a new task, such as fine-tuning a pretrained model by samples from a new task [21]. The meta-learning-based methods learn meta-knowledge of how to learn from a new task by meta-training episodes. Then the well-trained meta-knowledge is applied to a new task in meta-testing episodes for evaluation. UEFTC focuses on meta-learning-based methods. As a representative SOTA meta-learning-based method, FTC-DS [2] learns token embedding with the assistance of distributional signatures. Also mining assisted information, Geng et al. [17] dynamically update knowledge from base classes by a memory module. Meta-level attention is learned in LEA [28] based on pre-trained language models. In addition, MLADA [23] uses a generator and a discriminator to conduct adversarial learning for domain adaption in few-shot text classification. Since FTC-DS is a frequently used baseline and its attention-based token embedding is common among few-shot text classification, we use FTC-DS to study UEFTC.

Contrastive learning. Contrastive learning has been broadly applied in unsupervised representation learning [3, 7, 8, 19] by reducing the distance between positive pairs and enlarging the distance between negative pairs. Recently, contrastive learning has also been applied in supervised learning [35, 61] by positive and optional negative pairs, such as fine-tuning pre-trained language models [12, 20]. Many contrastive learning models use the detach, predictor, or intersection comparison components [3, 9, 19, 20, 42, 43, 54, 62]. How contrastive learning is used in supervised learning can be further divided into two categories: using negative samples and using no negative samples. For instance, Yeh et al. [65] propose decoupled contrastive loss and Wang and Qi [58] minimize the divergence between the weak and the strong augmented samples, they both need negative samples in the model training. As an example of using no negative samples, SimSiam [9] finds that using a Siamese net with detach operation achieves similar results by only a single sample in each update. Due to the few-support-sample limitation in UEFTC, contrastive learning using no negative samples [9] is more suitable, which further reduces the burden of the required sample size. Besides, unlike previous contrastive learning models [7, 9, 19, 58, 65], which only have equal relations between the augmented samples, our proposed uncertainty relations have additional unequal relations. Their equal relations are samples having the same (=) or different (≠) classes, but our unequal relations are larger (>) or smaller (<) uncertainty relations between the sample uncertainty.

3 PRELIMINARY KNOWLEDGE

3.1 UEFTC Task Settings

Problem Statement. As shown in Fig. 1, besides classifying texts like a few-shot text classification model, UEFTC additionally estimates the uncertainty scores for the classification results so that we can decide whether to trust the model prediction or not. A UEFTC model aims to learn how to acquire knowledge from training samples among training classes L^{Tr} during the meta-training. Then, given new classes L^{Te} during meta-testing, which are disjoint from L^{Tr}, a well-trained UEFTC model can quickly learn how to predict classes of testing samples among L^{Te} and their uncertainty scores. A better UEFTC model not only achieves higher classification performance but also gives higher uncertainty scores for misclassified results and lower uncertainty scores for correct results.

Meta-training. To meta-train a few-shot text classification model or a UEFTC model Θ, we create training episodes, shown as Fig. 1. Among a p-shot k-way setting, each training episode is built by randomly sampling k classes from L^{Tr}. From each of these k classes, we randomly sample p samples as a training set and m samples as a testing set. We update Θ based on loss over these testing samples in each training episode. We also repeat this model update among other p-shot k-way training episodes. In [2, 57], the training set of a training/testing episode is known as the support set, and its testing set is called the query set. Given the support set, we call Θ as a p-shot k-way few-shot model. According to different tasks, Θ can be a p-shot k-way few-shot text classification model or a p-shot k-way UEFTC model.

Meta-testing. To meta-test Θ, we use the same p-shot k-way setting to extract testing episodes from L^{Te}, shown as Fig. 1. For each testing episode, we use $p \times k$ support samples to update the model, which is further evaluated by the $m \times k$ query samples. A few-shot text classification model is evaluated by the classification accuracy on testing query sets. Besides the classification performance, we further evaluate a UEFTC model by the accuracy of estimated uncertainty scores of classification results on testing query sets. We aim to learn a UEFTC model predicting more accurate uncertainty scores for the classification results of testing query samples.

3.2 Few-Shot Text Classification Model

A few-shot text classification model aims at predicting unseen classes of query samples in testing episodes. Each episode [3] samples a support set and a query set randomly. We build CLUR on a SOTA model called FTC-DS [2], which uses attention-based tokens that are common-used in few-shot text classification [16, 18, 32, 56]. In each episode, FTC-DS first gets token sequences in discrete numbers for both support and query samples. It then applies a frozen pre-trained token embedding to represent the support and query texts. Thus, a sequence of vectors $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_n]$ for a text is obtained by the embedding, where $\mathbf{z}_i$ is the embedding of the i-th word. For a text embedding $\mathbf{x}$, FTC-DS sums each token embedding $\mathbf{z}$ as,

$$\mathbf{x} = \sum_i \alpha_i \cdot \mathbf{z}_i, \tag{1}$$

where $\alpha = [\alpha_0, \alpha_1, ..., \alpha_n]$ is the attention to learn. Then, FTC-DS gets a projection matrix $\mathbf{W}$, which is the solution of a regression loss $L = ||\mathbf{X}_S\mathbf{W} - \mathbf{Y}_S||_F^2 + \lambda||\mathbf{W}||_F^2$. To solve it, we have $\mathbf{W} = \mathbf{X}_S^T(\mathbf{X}_S\mathbf{X}_S^T + \lambda\mathbf{I})^{-1}\mathbf{Y}_S$, where $\mathbf{X}_S$ and $\mathbf{Y}_S$ are the text embeddings and labels of all support samples in an episode, $\mathbf{I}$ is an identity matrix, while λ is a scalar to be learned. Finally, FTC-DS predicts semantic vectors for a query set by a projector g as,

$$\widetilde{\mathbf{Y}}_Q = g(\mathbf{X}_Q) = a\mathbf{X}_Q\mathbf{W} + b \tag{2}$$

where $\{\cdot\}_Q$ has similar mean to $\{\cdot\}_S$, but $\{\cdot\}_Q$ refers to query samples. The model output $\widetilde{\mathbf{Y}}_Q \in \mathbb{R}^{m\times k}$ is a sequence of semantic vectors in an episode. The a and b in Eq. 2 are learnable scalars. In each training episode, FTC-DS updates α, λ, a, b by cross-entropy loss L_{CE} between model output $\widetilde{\mathbf{Y}}_Q$ and query set labels $\mathbf{Y}_Q$. In each testing episode, FTC-DS uses Eq. 2 to get $\widetilde{\mathbf{Y}}_Q$ for model evaluation.

3.3 SimSiam

SimSiam [9] is a SOTA contrastive learning model that only uses positive pairs for representation learning. The other SOTA contrastive learning models additionally require numerous negative pairs or large batch sizes, while SimSiam does not. Therefore, SimSiam is more suitable for UEFTC than other SOTA models as it requires fewer training samples. Plus, SimSiam also uses the three common-used contrastive learning components. As a result, our usage of data augmentation in CLUR is motivated by SimSiam. We briefly introduce SimSiam and the three components.

Given a sample $\mathbf{t}$, SimSiam augments it twice. The augmented samples $\mathbf{t}_1$ and $\mathbf{t}_2$ from $\mathbf{t}$ are input to a projector, which outputs projections $\widetilde{\mathbf{y}}_1 \in \mathbb{R}^k$ and $\widetilde{\mathbf{y}}_2 \in \mathbb{R}^k$ respectively, where k is class number. Finally, the projections $\widetilde{\mathbf{y}}_1$ and $\widetilde{\mathbf{y}}_2$ are input to a predictor, which outputs predictions $\hat{\mathbf{y}}_1 \in \mathbb{R}^k$ and $\hat{\mathbf{y}}_2 \in \mathbb{R}^k$ respectively. Its loss function is,

$$L_{SimSiam} = D[\widehat{\mathbf{y}}_1, o(\widetilde{\mathbf{y}}_2)] + D[\widehat{\mathbf{y}}_2, o(\widetilde{\mathbf{y}}_1)] \tag{3}$$

where D is cosine similarity and o is the *detach* (DT) operation to stop the gradient [9, 19, 54, 62]. D scales between $\widetilde{\mathbf{y}}$ and $\widehat{\mathbf{y}}$, which is an *intersection comparison* (IT) [9, 19, 42, 43]. Projections $\widetilde{\mathbf{y}}$ can be used for model inference, but many contrastive learning models extra design the *predictor* (PD) [3, 8, 19, 20] and use its predictions $\hat{\mathbf{y}}$ for inference. Since DT, IT, and PD are common in contrastive learning, we investigate whether they improve CLUR. Our findings can benefit the design of future UEFTC models with data augmentation.

4 OUR MODEL: CLUR

4.1 Overview Of CLUR

The upper panel of Fig. 2(L) shows the training process of our UEFTC model, Contrastive Learning from Uncertainty Relations (CLUR). CLUR is a pseudo-Siamese net [63], having two identical submodels with the same structure but different weights. In the first row (first submodel) of the upper panel, we augment the texts from a support set and a query set in each training episode. We then get the text embeddings for the support and query sets by the "embedding1" module. The query text embeddings are input to a projector g_1 to get their projections. These query text projections are further input to a predictor f_1 for their predictions. Similarly, the other projections and predictions for the same query set are obtained by the second submodel, shown in the second row with blue arrows in Fig. 2(L). Moreover, we design four choices of loss modules to train CLUR, shown in the bottom panel of Fig. 2(L). The four choices of loss modules verify whether DT, IT, and PD (defined in Sec. 3.3) help improve UEFTC. During the testing process, CLUR only uses the first submodel and skips the "augmentation1" module to get query text classification and uncertainty estimation results.

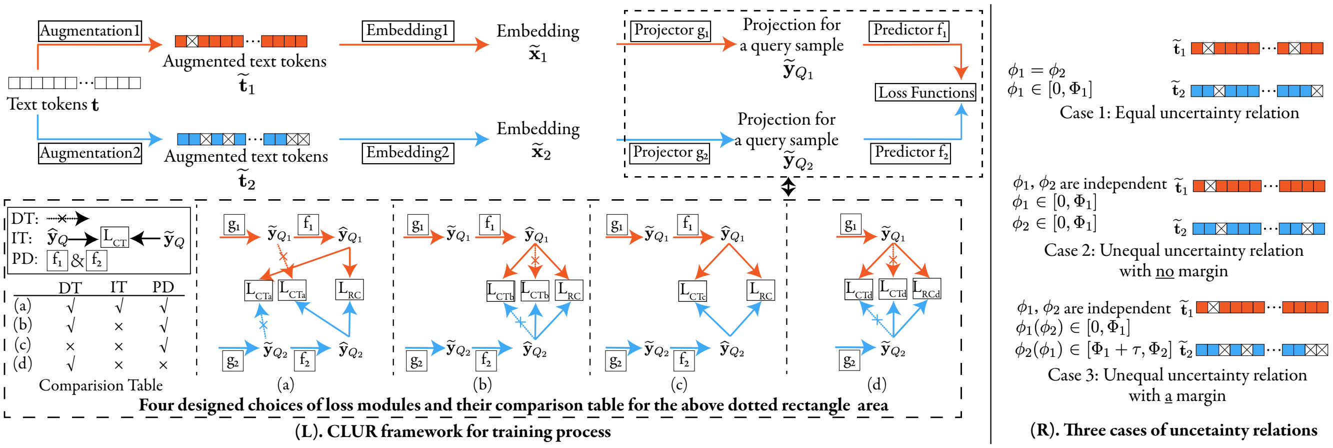
4.2 CLUR Training: Uncertainty Relations

As we don't know the true pseudo uncertainty scores, manually setting them can be inaccurate. Instead, CLUR self-adaptively learns the pseudo uncertainty scores via our uncertainty relations. Below, we introduce the augmentation module, which generates the uncertainty relations.

Augmentation module. Our data augmentation method is token-mask [64], which randomly masks the tokens in a text. We choose this data augmentation method, as it only needs one original sample, satisfying UEFTC's few-support-sample limitation.

Uncertainty Relations. Given an n-word text from either a support set or a query set, we have its token vector $\mathbf{t} \in \mathbb{R}^n$ in discrete

[3] An "episode" without defining a training or testing episode applies to both training and testing episodes. And a "sample" without defining a support or query sample is also applicable to both.


Augmentation1
Augmented text tokens
$\widetilde{\mathbf{t}}_1$
Text tokens t
Augmentation2
Augmented text tokens
$\widetilde{\mathbf{t}}_2$
Embedding1
Embedding
$\widetilde{\mathbf{x}}_1$
Embedding2
Embedding
$\widetilde{\mathbf{x}}_2$
Projector g1
Projection for a query sample
$\widetilde{\mathbf{y}}_{Q_1}$
Predictor f1
Loss Functions
Projector g2
Projection for a query sample
$\widetilde{\mathbf{y}}_{Q_2}$
Predictor f2
DT:
IT: $\widetilde{\mathbf{y}}_Q$ L_{CT} $\widetilde{\mathbf{y}}_Q$
PD: f1 & f2
DT IT PD
(a) √ √ √
(b) √ × √
(c) × × √
(d) √ × ×
Comparision Table
g1 f1 L_{CTa} L_{CTa} L_{RC} g2 f2
(a)
g1 f1 L_{CTb} L_{CTb} L_{RC} g2 f2
(b)
g1 f1 L_{CTc} L_{RC} g2 f2
(c)
g1 L_{CTd} L_{CTd} L_{RCd} g2
(d)
Four designed choices of loss modules and their comparison table for the above dotted rectangle area
$\phi_1 = \phi_2$
$\phi_1 \in [0, \Phi_1]$
Case 1: Equal uncertainty relation
ϕ_1, ϕ_2 are independent
$\phi_1 \in [0, \Phi_1]$
$\phi_2 \in [0, \Phi_1]$
Case 2: Unequal uncertainty relation with no margin
ϕ_1, ϕ_2 are independent
$\phi_1(\phi_2) \in [0, \Phi_1]$
$\phi_2(\phi_1) \in [\Phi_1 + \tau, \Phi_2]$
Case 3: Unequal uncertainty relation with a margin

(L). CLUR framework for training process

(R). Three cases of uncetainty relations

Figure 2: The left diagram (Fig. 2(L)) shows the training process of CLUR. The right diagram (Fig. 2(R)) shows three cases of uncertainty relations. Red and blue represent the data related to the first and second submodels respectively. The bottom dotted rectangle in Fig. 2(L) details four choices of loss modules and their comparison table, where "✓" ("×") means the design is used (unused) in the respective loss module. Fig. 2(R) illustrates our three cases of uncertainty relations by an example, where the example has $\Phi_1 = \tau = \Phi_2 = \frac{2}{n}$. Case 1 claims $\widetilde{\mathbf{t}}_1$ and $\widetilde{\mathbf{t}}_2$ have the same uncertainty; case 2 and case 3 both claim that $\widetilde{\mathbf{t}}_1$ has smaller uncertainty than $\widetilde{\mathbf{t}}_2$, where we verify the case 3 has more accurate pseudo uncertainty relations due to τ by our results in Sec. 5.2 and analysis in Sec. A.2.1. During the testing process, we only use the first submodel and skip the "augmentation1" module to get the classification results and their uncertainty scores.

numbers. We randomly mask $\mathbf{t}$ twice and obtain two different augmented token vectors $\widetilde{\mathbf{t}}_1 \in \mathbb{R}^n$ and $\widetilde{\mathbf{t}}_2 \in \mathbb{R}^n$ as below,

$$\begin{aligned} \widetilde{\mathbf{t}}_1 &= \mathbf{t} \cdot \mathbf{m}_{\phi_1} \\ \widetilde{\mathbf{t}}_2 &= \mathbf{t} \cdot \mathbf{m}_{\phi_2} \end{aligned} \tag{4}$$

where $\mathbf{m}_{\phi_1}$ and $\mathbf{m}_{\phi_2}$ are binary vectors to randomly mask $\mathbf{t}$ by ratios of ϕ_1 and ϕ_2, respectively. The ratios ϕ_1 and ϕ_2 are random numbers between 0 and 1 for each sample in each epoch. The different cases of ϕ_1 and ϕ_2 generate various uncertainty relations. We list three cases below, where their examples are shown in Fig. 2(R).

Case 1: Equal uncertainty relation. $\phi_1 \in [0, \Phi_1]$ and $\phi_1 = \phi_2$, where Φ_1 is a boundary of ϕ_1. In case 1, binary-mask vectors $\mathbf{m}_{\phi_1}$ and $\mathbf{m}_{\phi_2}$ have the same numbers of 0s and 1s but in random order. Since $\phi_1 = \phi_2$ in case 1, a pair of augmented samples from a text has the same numbers of the masked tokens. Due to their same numbers of masked tokens, case 1 assumes $\widetilde{\mathbf{t}}_1$ and $\widetilde{\mathbf{t}}_2$ have the same uncertainty. Thus, data augmentation in case 1 generates equal uncertainty relations.

Case 2: Unequal uncertainty relation without a margin. In case 2, $\phi_1 \in [0, \Phi_1]$, $\phi_2 \in [0, \Phi_1]$, ϕ_1 and ϕ_2 are independent. Then, the number and order of 0s and 1s in $\mathbf{m}_{\phi_1}$ and $\mathbf{m}_{\phi_2}$ can both be different. Since a text with more masked tokens is harder for a model to predict, the model prediction is more uncertain. For a pair of augmented texts, case 2 regards an augmented text with more masked tokens as more uncertain than the other augmented text with fewer masked tokens. Compared to case 1, case 2 has more data diversity to provide more information. This is because case 1 limits $\phi_1 = \phi_2$, but ϕ_1 and ϕ_2 in case 2 are independent, with $\frac{1}{n}$ probability that $\phi_1 = \phi_2$ and $\frac{n-1}{n}$ probability that $\phi_1 \neq \phi_2$.

Case 3: Unequal uncertainty relation with a margin. Case 3 is designed to solve the issues in cases 1 and 2. Specifically, case 1 assumes a pair of texts with the same number of masked tokens has equal uncertainty. But each token has a different contribution to the text classification. Second, even though we have $\phi_1 < \phi_2$ or $\phi_1 > \phi_2$ in case 2, the divergence between the numbers of masked tokens might be too small to crucially impact the uncertainty relations. Though each token contributes differently to text semantics, a larger difference in the masked token numbers leads to a more accurate pseudo uncertainty relation [4]. Thus, case 3 sets a margin τ to enlarge the divergence in the masked token numbers among a pair of augmented samples. Specifically, case 3 has a 50% chance that $\phi_1(\phi_2) \in [0, \Phi_1]$ and $\phi_2(\phi_1) \in [\Phi_1 + \tau, \Phi_2]$, where Φ_2 is a boundary and the other 50% chance is shown in the brackets. In case 3, one augmented sample has at most $\Phi_1 \times n$ masked tokens, and the other one has at least $(\Phi_1 + \tau) \times n$ masked tokens.

In the three cases, we set $\Phi_1 \in (0, 0.5]$ and $\Phi_2 \in (0, 0.5]$ to avoid losing much text context. Any above uncertainty relations are then used to learn pseudo uncertainty scores described in Sec. 4.4.

Advantage of uncertainty relations. In short, it is easier to manually set pseudo uncertainty relations than to manually set pseudo uncertainty scores. Concretely, the uncertainty relations have only three possible values (=, >, <), but the uncertainty scores have countless possible values (any number>0). Further, when the divergence of a pair of augmented samples is enlarged (such as enlarged by τ in case 3), the uncertainty relations are more explicit to us, but the knowledge about pseudo uncertainty scores does not increase. Thus, the much fewer possible values and more observable change in setting uncertainty relations make it easier than setting pseudo uncertainty scores.

[4] Detailed analysis is in Sec. A.2.1

4.3 CLUR Training: General Modules

Embedding module. As shown in the upper panel of Fig. 2(L), after getting a pair of augmented token vectors $\widetilde{\mathbf{t}}_1$ and $\widetilde{\mathbf{t}}_2$ by a chosen case for either a support or query text in training episodes, we use the frozen pre-trained word embedding to get their token embeddings $\widetilde{\mathbf{Z}}_1 \in \mathbb{R}^{n\times u}$ and $\widetilde{\mathbf{Z}}_2 \in \mathbb{R}^{n\times u}$, where u is the dimension of word embedding. Similar to Eq. 1, we then accumulate each token embedding in $\widetilde{\mathbf{Z}}_1$ by learnable attentions $\widetilde{\alpha}_1 = [\widetilde{\alpha}_{11}, \widetilde{\alpha}_{12}, ..., \widetilde{\alpha}_{1n}]$ to get the text embedding $\widetilde{\mathbf{x}}_1 \in \mathbb{R}^u$, where $\widetilde{\mathbf{x}}_1 = \sum_{i=1}^{n} \widetilde{\alpha}_{1i} \cdot \widetilde{\mathbf{z}}_{1i}$. Similarly, we get another text embedding $\widetilde{\mathbf{x}}_2 \in \mathbb{R}^u$ from $\widetilde{\mathbf{Z}}_2$.

Projector module. We then use a projector g_1 like Eq. 2 to get our projection $\widetilde{\mathbf{y}}_{Q_1} \in \mathbb{R}^k$ for a query sample, by

$$\widetilde{\mathbf{y}}_{Q_1} = g_1(\widetilde{\mathbf{x}}_{1Q}) = a_1\widetilde{\mathbf{x}}_{1Q}\mathbf{W}_1 + b_1 \tag{5}$$

where $\widetilde{\mathbf{x}}_{1Q}$ is an augmented query text embedding vector from the first submodel, a_1 and b_1 are the learnable parameters. For $\mathbf{W}_1$, it is calculated as,

$$\mathbf{W}_1 = \widetilde{\mathbf{X}}_{1S}^T(\widetilde{\mathbf{X}}_{1S}\widetilde{\mathbf{X}}_{1S}^T + \lambda_1\mathbf{I})^{-1}\mathbf{Y}_S \tag{6}$$

where $\widetilde{\mathbf{X}}_{1S}$ is the augmented support texts' embedding tensor from the first submodel. λ_1 is a learnable parameter. The $\mathbf{W}_1$ in Eq. 6 is the solution of a regression loss, $L_1 = ||\widetilde{\mathbf{X}}_{1S}\mathbf{W}_1 - \mathbf{Y}_S||_F^2 + \lambda||\mathbf{W}_1||_F^2$. Similar to Eq. 5 and Eq. 6, we get $\widetilde{\mathbf{y}}_{Q_2} \in \mathbb{R}^k$ by g_2, which has the same structure as g_1 but with different parameters.

4.4 CLUR Training: Explored Structures

Predictor module. The uncertainty relations are obtained from data augmentation. CLUR adopts contrastive learning in which the usage of data augmentation has been continuously optimized. Like the contrastive learning methods commonly do, the projection $\widetilde{\mathbf{y}}_{Q_1}$ is then input to a predictor f_1. The output of f_1 is the prediction of CLUR, $\widehat{\mathbf{y}}_{Q_1} \in \mathbb{R}^k$. Similarly, we get $\widehat{\mathbf{y}}_{Q_2} \in \mathbb{R}^k$ by f_2.

Loss Modules. To investigate the effects of three common-used contrastive learning modules (detach, intersection comparison, and predictor introduced in Sec. 3.3) in CLUR, we design four loss modules based on the projections $\widetilde{\mathbf{y}}_{Q_1}$, $\widetilde{\mathbf{y}}_{Q_2}$ and predictions $\widehat{\mathbf{y}}_{Q_1}$, $\widehat{\mathbf{y}}_{Q_2}$. The bottom panel in Fig. 2(L) shows the designed loss module choices. Only contrastive loss L_{CT_a} (case 1 in Sec. 4.2, Eq. 8) in loss choice (a) is similar to SimSiam. All the other loss choices are our original designs. Loss (b) has the best performance among the four loss module choices in UEFTC by removing the component "intersection comparison". (c) shows the performance of "detach". And (d) verifies the effectiveness of the "predictor".

Loss module (a). We first design a revised cross-entropy loss L_{RC} to keep classification performance and calibrate the prediction $\hat{\mathbf{y}}_Q$ for uncertainty estimation. In a training episode, a query text with one-hot label $\mathbf{y}_Q \in \mathbb{R}^k$ has its L_{RC} as,

$$\begin{aligned} L_{RC} = &\max\{L_{CE}(\widehat{\mathbf{y}}_{Q_1}, \mathbf{y}_Q) + \log(\beta), 0\}+ \\ &\max\{L_{CE}(\widehat{\mathbf{y}}_{Q_2}, \mathbf{y}_Q) + \log(\beta), 0\} \end{aligned} \tag{7}$$

where L_{CE} is the traditional cross-entropy loss between the prediction $\widehat{\mathbf{y}}_Q$ and one-hot label $\mathbf{y}_Q$. We add $\log(\beta)$, $\beta \in [0.5, 1)$ to each L_{CE}, which has a penalty of 0 for L_{RC} once the probability for the correct class is above β. With our L_{RC}, the probability for the correct class in $\widehat{\mathbf{y}}_Q$ is not always close to 1, but has feasible solutions in a larger range $[\beta, 1)$, this is further explained in Sec. A.2.2. With a larger feasible solution range $[\beta, 1)$, CLUR can learn different uncertainty scores for different samples. This is because if all predictions $\widehat{\mathbf{y}}_Q$ are always close to one-hot labels, the uncertainty scores (e.g., reciprocal of winning scores in Sec. 4.5) of the predictions are almost the same. Second, we design a contrastive loss L_{CT_a} for a pair of augmented samples with an equal uncertainty relation,

$$L_{CT_a} = D[\widehat{\mathbf{y}}_{Q_1}, o(\widetilde{\mathbf{y}}_{Q_2})] + D[\widehat{\mathbf{y}}_{Q_2}, o(\widetilde{\mathbf{y}}_{Q_1})] \tag{8}$$

where D is cosine similarity and o is the detach operation to stop the gradient, similar to Eq. 3 in SimSiam. Eq. 8 scales between $\widetilde{\mathbf{y}}$ and $\widehat{\mathbf{y}}$, an intersection comparison. The total loss of module (a) is $L_{SUM_a} = L_{RC} + \gamma L_{CT_a}$, where γ is a constant.

Loss module (b). This performs best among the four loss choices in UEFTC by ϕ_1 and ϕ_2 in case 3. It also has two components, the same L_{RC} as Eq. 7, and its contrastive loss L_{CT_b}. Though the L_{CT_a} has the interaction comparison between projection $\widetilde{\mathbf{y}}_{Q_1}$ and prediction $\widehat{\mathbf{y}}_{Q_1}$, it is hard to explain why we should compare these two, and its effect is shown to be inconsistent by our experiments (Sec. 5.2.2). As a result, we only compare the uncertainty relations among the predictions $\widehat{\mathbf{y}}_Q$ in (b). For a pair of augmented samples with an equal uncertainty relation (case 1), we have its L_{CT_b} as,

$$L_{CT_b} = D[\widehat{\mathbf{y}}_{Q_1}, o(\widehat{\mathbf{y}}_{Q_2})] + D[\widehat{\mathbf{y}}_{Q_2}, o(\widehat{\mathbf{y}}_{Q_1})] \tag{9}$$

For the augmentations with an unequal uncertainty relation (case 2 or 3), different from Eq. 3 in SimSiam, we propose its L_{CT_b} as,

$$\begin{aligned} L_{CT_b} = &max\{[H(\widehat{\mathbf{y}}_{Q_1}) - H(o(\widehat{\mathbf{y}}_{Q_2}))] \times (\phi_2 - \phi_1), 0\} \\ &+ max\{[H(\widehat{\mathbf{y}}_{Q_2}) - H(o(\widehat{\mathbf{y}}_{Q_1}))] \times (\phi_1 - \phi_2), 0\} \end{aligned} \tag{10}$$

where $H(\widehat{\mathbf{y}}) = -\sum \widehat{y}_i \log(\widehat{y}_i)$ calculates entropy. A larger entropy means more uncertainty. In the first item of Eq. 10, the $(\phi_2 - \phi_1)$ calculates our pseudo unequal relations (>, <). The $[H(\widehat{\mathbf{y}}_{Q_1}) - H(o(\widehat{\mathbf{y}}_{Q_2}))]$ calculates the model predicted unequal relations. If the predicted unequal relations and our pseudo unequal relations were the same, the predicted uncertainty scores from CLUR would be adaptive to our pseudo unequal relations, and the loss would be a constant 0 with no penalty. But if the predicted unequal relations and our pseudo unequal relations were different, the predicted uncertainty scores from CLUR would not be adaptive to our pseudo unequal relations, and there would be a positive loss as a penalty. The total loss is $L_{SUM_b} = L_{RC} + \gamma L_{CT_b}$.

Loss modules (c) & (d). Compared with (b), (c) shows the effectiveness of detach in CLUR, where L_{CT_c} only removes detach operation o; (d) shows the performance of predictors in CLUR by removing the predictors and learning via projections. Their loss functions are shown in the Sec. A.2.2.

4.5 CLUR Inference: Uncertainty Score

During the testing process, its support and query samples from a testing episode all go through the first submodel by skipping the augmentation. Thus, our testing process is not affected by the augmentation, but only uses the knowledge of learning text classification and uncertainty estimation that is learned in the training process. Then, like [25, 50], we calculate uncertainty score $\varGamma$ by the reciprocal of maximum probability in $\widehat{\mathbf{y}}_{Q_1}$, that is $\varGamma = \frac{1}{\max(\widehat{\mathbf{y}}_{Q_1})}$.

Generalization of CLUR. When applying uncertainty relations to other few-shot models, the two modules introduced in Sec. 4.3 are replaceable by other few-shot models to get their respective sample embeddings $\widetilde{\mathrm{X}}$ and query sample projects $\widetilde{\mathrm{Y}}_Q$. Then, the data augmentation for uncertainty relations (Sec. 4.2) and their respective loss functions (Sec. 4.4) are still applicable. We verify its generalization in Sec. A.3.2.

5 EXPERIMENTS

5.1 Experimental Setup

Datasets. We use four real-world datasets: (1) 20 Newsgroups (**20News**) [37] includes 20 news categories with 18,828 documents in it. (2) Amazon Reviews (**Amazon**) [46] is a set of reviews [26]. We use its subset provided by [2], which has 1000 reviews from each category. (3) HuffPost headlines (**HuffPost**) provides news headlines from HuffPost between 2012 and 2018 [47]. It has 36900 headlines with 41 classes. They are shorter and less grammatical than formal sentences. (4) **RCV1** collects Reuters articles from 1996 to 1997 [41]. We use its 71 second-level topics as labels, and discard its multi-label articles. Each dataset is split in the same way as [2].

Metrics. To evaluate the performance of UEFTC, we use three metrics. The first two are the area under the receiver operating characteristic curve (**AUROC**) and the area under the precision-recall curve (**AUPR**), which are broadly applied in uncertainty estimation [27, 29, 45, 69]. Higher AUROC and AUPR both mean a higher probability that a true prediction has a lower uncertainty score than a false prediction. Besides, to simulate the performance improvement of uncertainty scores with human involvement, we scale classification accuracy in different eliminated ratios [25, 68]. Concretely, for a testing episode with N query samples and eliminated ratio r, the most uncertain predictions in size of $N \times r$ are set as true. The more accurate the uncertainty scores we obtain, the more misclassified predictions will be set as true predictions under the same r, resulting in a larger F1 score. The F1 score under 0% eliminated ratio is the model classification performance.

Baselines and Ablation Settings. We use six baselines. **FTC-DS** [2] is the SOTA few-shot text classification model described in Sec. 3.2. To ensure fairness, all other baselines are also built on FTC-DS, like CLUR. [25, 68] have the same tasks as ours, but they use numerous training samples. Zhang et al. [68] propose two methods: Dropout-Entropy (**DE**) is a dropout-based model. **DE+Metric** additionally uses metric learning. There are two other methods in He et al. [25] applicable to UEFTC: **MSD1** uses mix-up to manually set pseudo uncertainty scores; **MSD2** adds self-ensembling components to MSD1. We refer to our CLUR with loss module (a) using equal uncertainty relation as **SimSiam** [9], since it uses similar key structures and key loss L_{CT_a} (Eq. 8) as SimSiam (Eq. 3).

For the ablation studies, we design five comparisons listed in Tab. 4. They are in different cases and structures to compare different designs: detach (**DT**), predictor (**PD**), and intersection comparison (**IT**) between the projection and prediction (described in Sec. 3.3). We use CLUR-{·}-{★} to represent a CLUR using the loss module {·} in case {★} described in Sec. 4.4 and 4.2 respectively.

Implementation Details. We use fastText [33] as the word embedding for our experiments by default. Besides fastText, we also test BERT [11] word embedding for 5-way 1-shot on 20News. Our parameter settings are listed in Sec. A.3.1 and Tab. 7.

5.2 Experimental Results

5.2.1 Comparison With Baselines. Tab. 1 and 3 report the CLUR improvement in UEFTC using fastText in the 5-way 1-shot and 5-way 5-shot settings respectively. We repeat the testing process 30 times with the same dropout rate. And we calculate the mean and standard deviation for each metric, which are reported in the tables. From the two tables, we discuss below questions:

1. Are the learned pseudo uncertainty scores from uncertainty relations better than manual setting ones? Yes, CLUR-b in case 3 performs better than MSD1 and MSD2, which both manually set pseudo uncertainty scores, such as 4.52% AUPR improvement than MSD1 in the 5-way 1-shot setting on RCV1 in Tab. 1, and 1.97% AUROC improvement than MSD2 in the 5-way 5-shot setting on 20News in Tab. 3. Concretely, MSD1 and MSD2 both use the mix-up to augment the texts and then manually set the mix-up coefficient as pseudo uncertainty scores. Compared with them, CLUR learns the pseudo uncertainty scores by Eq. 10, instead of the manual setting. Though MSD1 has higher F1 scores in the eliminated ratios (0%-30%) than CLUR in 5-way 5-shot on HuffPost, CLUR surpasses MSD1 in AUROC and AUPR in the same setting. It means that CLUR predicts more accurate uncertainty scores in total. Thus, learned pseudo uncertainty scores from uncertainty relations are more accurate than manual setting ones.

2. Is CLUR better than traditional uncertainty estimation methods (dropout, metric learning, self-ensemble, and pseudo-label-based methods) applicable to a few training samples? Yes, they are. In detail, DE uses dropout, and DE+Metrics additionally uses metric learning to reduce uncertainty. MSD1 manually sets pseudo uncertainty scores; MSD2 extra uses self-ensemble [51] to reduce uncertainty. In the two tables, CLUR beats them by its uncertainty relations and structure design in vast comparisons. For example, CLUR improves 4.39% and 4.08% F1 scores with a 10% eliminated ratio than DE and DE+Metric, respectively, in the 5-way 1-shot setting on 20News (Tab. 1). Therefore, CLUR beats traditional uncertainty estimation methods applicable to few-shot settings.

Below, we discuss the results using BERT embedding for 5-way 1-shot on 20News, shown in Tab. 2.

3. Is CLUR effective on BERT embeddings? Yes, our unequal uncertainty relation in case 3 and (b) loss module in CLUR perform better on 20News, using BERT embeddings. For example, CLUR-b-3 improves 2.42% AUPR than MSD1 in Tab. 2.

5.2.2 Ablation Study. Tab. 4 shows ablation study results. From the results, we conclude as below.

4. Which combo choices of loss modules and uncertainty relations perform better? CLUR using detach, predictor, no intersection comparison (L_{CT_b} in Eq. 10) in an uncertainty relation with a margin (case 3 on Sec. 4.2) performs better on almost all datasets. For example, CLUR-b-3 improves 2.80% AUROC than CLUR-c-2 in a 5-way 5-shot setting on Amazon (Tab. 4).

5. Among the three cases of uncertainty relations, which one performs better? Case 3 performs the best among the three cases. In detail, from the view of L_{CT_b} in Eq. 10, we compare CLUR-b-3 and CLUR-b-2. We see that CLUR-b-3 is better than CLUR-b-2. Thus, using a margin (case 3) performs better than without a margin (case 2). In addition, from the view of L_{CT_a}, we compare CLUR-a-2 and CLUR-a-1 (SimSiam). Since CLUR-a-2 performs better than

Table 1: Results of baselines and CLUR using fastText word embedding in 5-way 1-shot setting, where standard deviations are behind "±". More results are in Tab. 3.

Methods	Uncertainty Ratio (F1 Score, Eliminated Ratio)↑					AUROC ↑	AUPR↑
	0%	10%	20%	30%	40%		
20News in the 5-way 1-shot setting							
FTC-DS	47.56±1.56	55.76±1.38	62.92±1.25	69.86±1.11	75.77±1.04	68.17±2.15	68.20±1.29
DE	52.32±1.70	59.45±1.59	65.71±1.47	72.12±1.32	77.57±1.27	67.69±2.44	69.38±1.57
DE+Metric	52.33±1.61	59.63±1.44	65.73±1.36	72.04±1.26	77.61±1.15	68.02±2.38	69.44±1.45
MSD1	53.11±1.60	60.47±1.47	66.61±1.36	72.87±1.26	78.38±1.09	68.40±2.35	70.01±1.36
MSD2	52.54±1.32	60.09±1.19	66.54±1.10	72.59±1.04	77.96±0.93	68.49±1.91	69.78±1.01
SimSiam(CLUR-a-1)	53.30±1.57	60.63±1.43	66.86±1.32	73.19±1.23	78.59±1.16	68.74±2.29	70.89±1.36
CLUR-b-3	**54.53**±1.50	**62.06**±1.37	**68.29**±1.25	**74.59**±1.11	**80.02**±0.98	**70.50**±2.13	**73.71**±1.22
RCV1 in the 5-way 1-shot setting							
FTC-DS	51.32±1.64	59.71±1.49	66.16±1.33	72.83±1.23	78.65±1.12	70.48±2.32	73.99±1.22
DE	55.42±1.62	62.96±1.50	68.91±1.37	74.99±1.22	80.09±1.14	70.72±2.34	75.12±1.12
DE+Metric	54.89±1.68	62.50±1.52	68.41±1.34	74.59±1.25	79.78±1.20	70.61±2.46	74.51±1.24
MSD1	54.91±1.79	62.32±1.64	68.27±1.48	74.60±1.36	79.82±1.26	70.11±2.50	73.67±1.35
MSD2	55.54±1.65	62.96±1.50	68.91±1.39	75.18±1.30	80.39±1.17	71.12±2.37	75.34±1.23
SimSiam(CLUR-a-1)	54.12±1.97	61.66±1.79	67.98±1.67	74.47±1.49	79.71±1.38	71.10±2.73	74.24±1.56
CLUR-b-3	**55.89**±1.60	**63.48**±1.44	**69.47**±1.35	**75.62**±1.23	**80.91**±1.12	**72.31**±2.26	**77.00**±1.10
Amazon in the 5-way 1-shot setting							
FTC-DS	59.06±1.49	66.81±1.30	72.65±1.27	78.22±1.14	82.73±1.01	70.05±1.96	79.03±0.97
DE	59.87±1.94	66.91±1.79	72.60±1.65	78.25±1.49	83.10±1.38	70.34±2.62	78.48±1.62
DE+Metric	61.36±1.65	68.39±1.51	73.83±1.37	79.13±1.27	83.55±1.15	70.66±2.37	79.63±1.15
MSD1	61.30±1.74	68.08±1.60	73.60±1.47	78.99±1.36	83.48±1.24	70.00±2.55	78.41±1.38
MSD2	61.56±1.34	68.30±1.20	73.87±1.11	79.24±1.05	83.78±1.00	70.70±1.93	80.02±0.90
SimSiam(CLUR-a-1)	61.42±1.87	68.13±1.73	73.60±1.59	78.93±1.42	83.37±1.27	69.66±2.52	78.66±1.41
CLUR-b-3	**63.32**±1.38	**70.08**±1.26	**75.41**±1.17	**80.69**±1.04	**85.13**±0.92	**71.59**±1.93	**81.78**±0.89
HuffPost in the 5-way 1-shot setting							
FTC-DS	40.65±1.48	48.92±1.33	56.25±1.25	63.64±1.14	70.31±1.08	66.35±2.21	61.35±1.37
DE	42.46±1.63	50.15±1.51	56.87±1.40	64.02±1.35	70.34±1.24	64.72±2.50	58.82±1.72
DE+Metric	42.55±1.40	50.27±1.31	57.19±1.19	64.33±1.10	70.64±1.07	65.48±2.26	59.96±1.29
MSD1	43.25±1.23	50.91±1.16	57.64±1.09	64.63±1.03	70.70±0.99	65.09±1.98	60.59±1.18
MSD2	42.92±1.12	50.70±1.03	57.32±0.97	64.23±0.92	70.58±0.87	64.88±1.80	59.41±1.04
SimSiam(CLUR-a-1)	43.18±1.31	50.73±1.24	57.46±1.17	64.58±1.12	70.80±1.02	65.42±1.96	61.41±1.24
CLUR-b-3	**44.05**±1.62	**51.62**±1.48	**58.26**±1.38	**65.20**±1.27	**71.39**±1.17	**66.50**±2.43	**63.07**±1.53

Table 2: 5-way 1-shot using BERT word embedding on 20News.

ID	Methods	Classification F1 Score (r = 0%)↑	AUROC↑	AUPR↑
1	FTC-DS	38.92	64.54	58.58
2	DE	44.81	63.40	59.45
3	DE+Metric	45.16	63.66	59.84
4	MSD1	45.75	64.00	61.09
5	MSD2	45.47	63.18	59.19
6	SimSiam(CLUR-a-1)	44.40	63.82	59.30
7	CLUR-b-3	**46.54**	**64.78**	**62.57**

CLUR-a-1, we conclude that case 2 is better than case 1. Together, case 3 performs the best among the three cases.

6. Do detach, intersection and predictor (introduced in Sec. 3.3) improve uncertainty estimation? From Tab. 4, we can conclude below. (i) The detach is effective by comparing the CLUR-b-3 and CLUR-c-3. This is because the detach acts as a form of structural ensemble and helps reduce uncertainty. (ii) As for the intersection comparison between projections and predictions, it is inconsistently effective in UEFTC by finding the slight difference between CLUR-a-2 and CLUR-b-2. (iii) The predictor leads to the most obvious improvement by comparing CLUR-b-3 and CLUR-d-3. This is because the predictor provides more parameters and better handles classification and uncertainty estimation simultaneously.

Generalization to other few-shot models. We conducted a generalization analysis, as shown in Tab. 5 and discussed in Sec. A.3.2.

Experiment on a high-risk domain dataset. Besides the four commonly used datasets, we are also interested in exploring the effectiveness of CLUR on public high-risk datasets, such as healthcare. Therefore, we conducted experiments on a medical-domain dataset, as shown in Tab. 6 and discussed in Sec. A.3.3.

6 CONCLUSION

This paper proposes CLUR to improve Uncertainty Estimation for Few-shot Text Classification. CLUR, which is based on data ensemble and pseudo label, overcomes the unique challenge of having few support samples in UEFTC. CLUR achieves UEFTC by

Table 3: Results of baselines and CLUR using fastText word embedding in 5-way 5-shot setting, where standard deviations are behind "±". More results are in Tab. 1.

Methods	Uncertainty Ratio (F1 Score, Eliminated Ratio)↑					AUROC ↑	AUPR↑
	0%	10%	20%	30%	40%		
20News in the 5-way 5-shot setting							
FTC-DS	63.83±1.18	70.87±1.05	76.76±0.92	82.31±0.79	86.86±0.68	76.22±1.69	85.46±0.54
DE	65.95±1.27	72.56±1.08	77.62±1.00	82.56±0.93	86.69±0.88	74.15±2.09	84.10±0.84
DE+Metric	65.82±1.07	72.58±0.95	77.81±0.83	83.08±0.77	87.34±0.70	75.52±1.72	84.65±0.84
MSD1	65.97±1.20	72.61±1.04	77.88±0.93	82.93±0.87	87.09±0.86	73.98±2.00	83.15±1.04
MSD2	65.83±1.06	72.35±0.95	77.83±0.85	83.10±0.79	87.24±0.74	74.72±1.71	83.37±0.79
SimSiam(CLUR-a-1)	66.02±1.31	72.71±1.17	78.02±1.01	83.10±0.94	87.15±0.90	74.78±2.03	84.10±0.96
CLUR-b-3	**66.88**±1.16	**73.71**±0.99	**79.20**±0.85	**84.26**±0.78	**88.21**±0.76	**76.50**±1.77	**86.39**±0.57
RCV1 in the 5-way 5-shot setting							
FTC-DS	72.28±1.63	78.82±1.42	83.35±1.23	87.79±1.11	91.29±0.97	77.26±2.54	89.65±0.75
DE	73.13±1.51	79.48±1.25	84.64±1.10	89.18±0.97	92.16±0.90	**80.11**±2.39	91.34±0.65
DE+Metric	74.52±1.46	80.49±1.19	84.90±1.11	89.01±1.06	92.20±0.93	77.57±2.56	90.86±0.69
MSD1	74.05±1.55	80.51±1.35	85.42±1.16	89.76±1.11	92.65±0.99	79.53±2.46	90.98±0.78
MSD2	74.44±1.25	80.75±1.03	85.22±0.98	89.52±0.86	92.56±0.75	79.24±2.01	91.2±0.57
SimSiam(CLUR-a-1)	73.51±1.49	80.14±1.27	84.60±1.10	88.76±1.02	92.08±0.91	79.11±2.37	90.80±0.71
CLUR-b-3	**75.88**±1.37	**82.09**±1.23	**86.31**±1.11	**90.28**±0.98	**93.17**±0.83	79.65±2.29	**91.55**±0.65
Amazon in the 5-way 5-shot setting							
FTC-DS	81.23±1.05	86.75±0.83	90.62±0.70	93.67±0.61	95.81±0.54	81.00±1.75	94.66±0.32
DE	81.07±1.26	86.48±1.05	90.31±0.93	93.49±0.77	95.58±0.66	80.93±2.15	94.29±0.51
DE+Metric	81.05±1.23	86.54±1.04	90.33±0.89	93.36±0.77	95.46±0.67	80.72±2.14	94.17±0.52
MSD1	81.79±1.24	87.01±1.04	90.81±0.85	93.84±0.75	95.91±0.63	81.07±2.05	94.72±0.47
MSD2	81.06±1.09	86.44±0.91	90.20±0.79	93.24±0.69	95.36±0.62	80.12±1.90	94.08±0.44
SimSiam(CLUR-a-1)	80.75±1.33	86.26±1.16	90.02±0.99	92.98±0.83	95.09±0.75	79.73±2.24	93.66±0.56
CLUR-b-3	**81.95**±1.09	**87.37**±0.90	**91.49**±0.76	**94.47**±0.57	**96.21**±0.51	**82.35**±1.79	**95.16**±0.36
HuffPost in the 5-way 5-shot setting							
FTC-DS	62.28±0.92	69.44±0.87	75.70±0.76	81.60±0.69	86.23±0.63	75.82±1.29	84.06±0.52
DE	63.80±1.20	70.79±1.06	76.48±0.99	81.86±0.91	86.22±0.79	74.74±1.74	83.50±0.72
DE+Metric	63.58±1.27	70.45±1.14	76.31±1.03	81.75±0.86	86.01±0.84	74.72±1.79	83.42±0.79
MSD1	**64.11**±1.14	**71.16**±1.03	**76.83**±0.92	**82.09**±0.85	86.35±0.77	74.80±1.64	83.64±0.76
MSD2	63.58±0.98	70.43±0.88	76.19±0.82	81.51±0.78	85.90±0.71	74.28±1.44	83.16±0.60
SimSiam(CLUR-a-1)	63.67±1.29	70.39±1.15	75.87±1.07	81.25±0.96	85.71±0.89	73.74±1.84	82.87±0.82
CLUR-b-3	63.55±1.37	70.74±1.22	76.63±1.12	82.04±1.00	**86.48**±0.89	**75.74**±1.89	**84.10**±0.82

Table 4: Ablation study of CLUR using fastText word embedding in the 5-way 5-shot setting on Amazon dataset, where standard deviations are behind "±."

Methods	Detach	Intersection	Predictor	Uncertainty Ratio (F1 Score, Eliminated Ratio) ↑					AUROC ↑	AUPR ↑
				0%	10%	20%	30%	40%		
Amazon in the 5-way 5-shot setting										
CLUR-b-3	✓	×	✓	**81.95**±1.09	**87.37**±0.90	**91.49**±0.76	**94.47**±0.57	**96.21**±0.51	**82.35**±1.79	**95.16**±0.36
CLUR-c-3	×	×	✓	81.44±1.09	86.91±0.94	90.59±0.77	93.63±0.70	95.76±0.61	81.26±1.92	94.52±0.43
CLUR-d-3	✓	×	×	80.17±2.09	85.90±1.76	89.93±1.48	93.33±1.23	95.58±1.02	81.13±3.05	94.33±0.92
CLUR-a-2	✓	✓	✓	80.83±1.29	86.32±1.12	90.14±0.96	93.33±0.82	95.50±0.71	80.69±2.15	94.23±0.55
CLUR-b-2	✓	×	✓	80.59±1.23	86.11±1.06	90.00±0.91	93.25±0.80	95.42±0.70	80.79±2.07	94.17±0.52
CLUR-c-2	×	×	✓	80.90±1.19	86.31±1.01	90.05±0.84	93.08±0.75	95.20±0.66	80.11±2.05	93.91±0.48

self-adaptively learning pseudo uncertainty scores using our proposed uncertainty relations instead of manually setting the pseudo uncertainty scores. Moreover, we investigate the effects of three commonly used contrastive learning components in UEFTC and discover that only the detach and predictor benefit the model. CLUR can be optimized by removing the intersection comparison component in the contrastive learning model. Experiments on four datasets demonstrated that CLUR using unequal uncertainty relation with a margin obtained more accurate uncertainty scores.

7 ACKNOWLEDGEMENT

This research is supported in part by National Science Foundation (NSF) grants CNS-2141095 and NSF grant 2050727.

REFERENCES

[1] Javier Antorán, James Urquhart Allingham, and José Miguel Hernández-Lobato. 2020. Depth uncertainty in neural networks. *arXiv preprint arXiv:2006.08437* (2020).

[2] Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. 2020. Few-shot Text Classification with Distributional Signatures. In *International Conference on Learning Representations*.

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882* (2020).

[4] Bertrand Charpentier, Oliver Borchert, Daniel Zügner, Simon Geisler, and Stephan Günnemann. 2022. Natural Posterior Network: Deep Bayesian Uncertainty for Exponential Family Distributions. *ICRL* (2022).

[5] 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.

[6] Baixu Chen, Junguang Jiang, Ximei Wang, Jianmin Wang, and Mingsheng Long. 2022. Debiased pseudo labeling in self-training. *arXiv preprint arXiv:2202.07136* (2022).

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[8] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. 2020. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297* (2020).

[9] Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15750–15758.

[10] davidberenstein1957. 2023. *A public medical domain dataset*. https://huggingface.co/datasets/argilla/medical-domain

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[12] Bo Dong, Yiyi Wang, Hanbo Sun, Yunji Wang, Alireza Hashemi, and Zheng Du. 2022. CML: A contrastive meta learning method to estimate human label confidence scores and reduce data collection cost. In *Proceedings of The Fifth Workshop on e-Commerce and NLP (ECNLP 5)*. 35–43.

[13] Bo Dong, Yuhang Wu, Micheal Yeh, Yusan Lin, Yuzhong Chen, Hao Yang, Fei Wang, Wanxin Bai, Krupa Brahmkstri, Zhang Yimin, et al. 2022. Semi-supervised Context Discovery for Peer-Based Anomaly Detection in Multi-layer Networks. In *Information and Communications Security: 24th International Conference, ICICS 2022, Canterbury, UK, September 5–8, 2022, Proceedings*. Springer, 508–524.

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

[15] Mark Gales and Andrey Malinin. 2021. UNCERTAINTY ESTIMATION IN AUTOREGRESSIVE STRUCTURED PREDICTION. (2021).

[16] Yao Ge, Yuting Guo, Yuan-Chi Yang, Mohammed Ali Al-Garadi, and Abeed Sarker. 2022. Few-shot learning for medical text: A systematic review. *arXiv preprint arXiv:2204.14081* (2022).

[17] Ruiying Geng, Binhua Li, Yongbin Li, Jian Sun, and Xiaodan Zhu. 2020. Dynamic memory induction networks for few-shot text classification. *arXiv preprint arXiv:2005.05727* (2020).

[18] Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot text classification. *arXiv preprint arXiv:1902.10482* (2019).

[19] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. 2020. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733* (2020).

[20] Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. 2021. Supervised contrastive learning for pre-trained language model fine-tuning. *International Conference on Learning Representations* (2021).

[21] Aakriti Gupta, Kapil Thadani, and Neil O'Hare. 2020. Effective few-shot classification with transfer learning. In *Proceedings of the 28th International Conference on Computational Linguistics*. 1061–1066.

[22] Muhammad Hamisu and Ali Mansour. 2021. Detecting advance fee fraud using nlp bag of word model. In *2020 IEEE 2nd International Conference on Cyberspac (CYBER NIGERIA)*. IEEE, 94–97.

[23] Chengcheng Han, Zeqiu Fan, Dongxiang Zhang, Minghui Qiu, Ming Gao, and Aoying Zhou. 2021. Meta-learning adversarial domain adaptation network for few-shot text classification. *arXiv preprint arXiv:2107.12262* (2021).

[24] Jianfeng He, Julian Salazar, Kaisheng Yao, Haoqi Li, and Jinglun Cai. 2023. Zero-Shot End-to-End Spoken Language Understanding via Cross-Modal Selective Self-Training. *arXiv preprint arXiv:2305.12793* (2023).

[25] Jianfeng He, Xuchao Zhang, Shuo Lei, Zhiqian Chen, Fanglan Chen, Abdulaziz Alhamadani, Bei Xiao, and ChangTien Lu. 2020. Towards More Accurate Uncertainty Estimation In Text Classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 8362–8372.

[26] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. 507–517.

[27] 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).

[28] SK Hong and Tae Young Jang. 2022. LEA: Meta Knowledge-Driven Self-Attentive Document Embedding for Few-Shot Text Classification. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 99–106.

[29] Yibo Hu and Latifur Khan. 2021. Uncertainty-Aware Reliable Text Classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 628–636.

[30] Yibo Hu, Yuzhe Ou, Xujiang Zhao, Jin-Hee Cho, and Feng Chen. 2020. Multidimensional Uncertainty-Aware Evidential Neural Networks. *arXiv preprint arXiv:2012.13676* (2020).

[31] Wenming Jiang, Ying Zhao, Yihan Wu, and Haojia Zuo. 2022. Capturing Model Uncertainty with Data Augmentation in Deep Learning. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*. SIAM, 271–279.

[32] Xiang Jiang, Mohammad Havaei, Gabriel Chartrand, Hassan Chouaib, Thomas Vincent, Andrew Jesson, Nicolas Chapados, and Stan Matwin. 2018. On the importance of attention in meta-learning for few-shot text classification. *arXiv preprint arXiv:1806.00852* (2018).

[33] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651* (2016).

[34] Alex Kendall and Yarin Gal. 2017. What uncertainties do we need in bayesian deep learning for computer vision?. In *Advances in neural information processing systems*. 5574–5584.

[35] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362* (2020).

[36] Aaron Klein, Stefan Falkner, Jost Tobias Springenberg, and Frank Hutter. 2017. Learning curve prediction with Bayesian neural networks. *International Conference on Learning Representations* (2017).

[37] Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*. Elsevier, 331–339.

[38] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*. 7167–7177.

[39] Shuo Lei, Xuchao Zhang, Jianfeng He, Fanglan Chen, and Chang-Tien Lu. 2022. Uncertainty-Aware Cross-Lingual Transfer with Pseudo Partial Labels. In *Findings of the Association for Computational Linguistics: NAACL 2022*. 1987–1997.

[40] Shuo Lei, Xuchao Zhang, Jianfeng He, Fanglan Chen, and Chang-Tien Lu. 2023. TART: Improved Few-shot Text Classification Using Task-Adaptive Reference Transformation. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics*.

[41] David D Lewis, Yiming Yang, Tony Russell-Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research* 5, Apr (2004), 361–397.

[42] Siyuan Li, Zelin Zang, and Stan Z Li. 2022. Exploring Localization for Self-supervised Fine-grained Contrastive Learning. (2022).

[43] Ran Liu, Mehdi Azabou, Max Dabagia, Chi-Heng Lin, Mohammad Gheshlaghi Azar, Keith Hengen, Michal Valko, and Eva Dyer. 2021. Drop, swap, and generate: A self-supervised approach for generating neural activity. *Advances in neural information processing systems* 34 (2021), 10587–10599.

[44] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. 2019. A simple baseline for bayesian uncertainty in deep learning. *Advances in Neural Information Processing Systems* 32 (2019).

[45] Andrey Malinin and Mark Gales. 2018. Predictive uncertainty estimation via prior networks. *Advances in neural information processing systems* 31 (2018).

[46] Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*. 165–172.

[47] Rishabh Misra. 2018. News category dataset. (2018).

[48] Subhabrata Mukherjee and Ahmed Awadallah. 2020. Uncertainty-aware self-training for few-shot text classification. *Advances in Neural Information Processing Systems* 33 (2020).

[49] Kazuki Osawa, Siddharth Swaroop, Mohammad Emtiyaz E Khan, Anirudh Jain, Runa Eschenhagen, Richard E Turner, and Rio Yokota. 2019. Practical deep learning with Bayesian principles. *Advances in neural information processing systems* 32 (2019).

[50] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems* 32 (2019).

[51] Sungrae Park, JunKeon Park, Su-Jin Shin, and Il-Chul Moon. 2018. Adversarial dropout for supervised and semi-supervised learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[52] Kanil Patel, William Beluch, Dan Zhang, Michael Pfeiffer, and Bin Yang. 2021. On-manifold adversarial data augmentation improves uncertainty calibration. In *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 8029–8036.

[53] Hippolyt Ritter, Martin Kukla, Cheng Zhang, and Yingzhen Li. 2021. Sparse Uncertainty Representation in Deep Learning with Inducing Weights. *arXiv preprint arXiv:2105.14594* (2021).

[54] Xiaohui Song, Longtao Huang, Hui Xue, and Songlin Hu. 2022. Supervised prototypical contrastive learning for emotion recognition in conversation. *arXiv preprint arXiv:2210.08713* (2022).

[55] Maximilian Stadler, Bertrand Charpentier, Simon Geisler, Daniel Zügner, and Stephan Günnemann. 2021. Graph Posterior Network: Bayesian Predictive Uncertainty for Node Classification. *Advances in Neural Information Processing Systems* 34 (2021).

[56] Shengli Sun, Qingfeng Sun, Kevin Zhou, and Tengchao Lv. 2019. Hierarchical attention prototypical networks for few-shot text classification. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*. 476–485.

[57] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. *Advances in neural information processing systems* 29 (2016).

[58] Xiao Wang and Guo-Jun Qi. 2022. Contrastive learning with stronger augmentations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[59] Yaqing Wang, Subhabrata Mukherjee, Haoda Chu, Yuancheng Tu, Ming Wu, Jing Gao, and Ahmed Hassan Awadallah. 2021. Meta self-training for few-shot neural sequence labeling. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1737–1747.

[60] Yuchao Wang, Haochen Wang, Yujun Shen, Jingjing Fei, Wei Li, Guoqiang Jin, Liwei Wu, Rui Zhao, and Xinyi Le. 2022. Semi-supervised semantic segmentation using unreliable pseudo-labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4248–4257.

[61] Zhuoyi Wang, Yigong Wang, Bo Dong, Sahoo Pracheta, Kevin Hamlen, and Latifur Khan. 2020. Adaptive margin based deep adversarial metric learning. In *2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*. IEEE, 100–108.

[62] Xing Wu, Chaochen Gao, Zijia Lin, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2022. InfoCSE: Information-aggregated Contrastive Learning of Sentence Embeddings. *arXiv preprint arXiv:2210.06432* (2022).

[63] Congying Xia, Caiming Xiong, and Philip Yu. 2021. Pseudo siamese network for few-shot intent generation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2005–2009.

[64] Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. ConSERT: A Contrastive Framework for Self-Supervised Sentence Representation Transfer. *arXiv preprint arXiv:2105.11741* (2021).

[65] Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. 2022. Decoupled contrastive learning. In *European Conference on Computer Vision*. Springer, 668–684.

[66] Sheheryar Zaidi, Arber Zela, Thomas Elsken, Chris Holmes, Frank Hutter, and Yee Whye Teh. 2021. Neural Ensemble Search for Uncertainty Estimation and Dataset Shift. *Advances in Neural Information Processing Systems* 34 (2021).

[67] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412* (2017).

[68] Xuchao Zhang, Fanglan Chen, Chang-Tien Lu, and Naren Ramakrishnan. 2019. Mitigating Uncertainty in Document Classification. In *Proceedings of NAACL-HLT*. 3126–3136.

[69] Xujiang Zhao, Feng Chen, Shu Hu, and Jin-Hee Cho. 2020. Uncertainty aware semi-supervised learning on graph data. *Advances in Neural Information Processing Systems* 33 (2020), 12827–12836.

A APPENDIX

A.1 Preliminary Knowledge

A.1.1 MSD: pseudo uncertainty. MSD1 [25] uses the mix-up to augment data and simulate the generation of uncertainty. It has an augmented sample $\widetilde{\mathbf{x}}$ by two text embedding $\mathbf{x}_i$ and $\mathbf{x}_j$ with their respective labels $\mathbf{y}_i$ and $\mathbf{y}_j$ as below,

$$\widetilde{\mathbf{x}} = \vartheta \mathbf{x}_i + (1-\vartheta)\mathbf{x}_j, \widetilde{\mathbf{y}} = \vartheta \mathbf{y}_i + (1-\vartheta)\mathbf{y}_j \tag{11}$$

where ϑ is a random number ranging from Ω to 1. The Ω is set above 0.5. It then learns a KL divergence loss for the augmented data. It uses a basic way to measure uncertainty, which is the reciprocal of the maximum probability of a softmax vector. Thus, its pseudo uncertainty is $\frac{1}{\vartheta}$ because of $\widetilde{\mathbf{y}}$, which is used as the pseudo uncertainty score to train an uncertainty estimation model. However, the manual-set pseudo uncertainty scores are inaccurate, because we have implicit knowledge of the pseudo uncertainty scores.

A.2 Model

A.2.1 Analysis Of Case 3 in Sec. 4.2. Assumption: We assume that the text semantics is not related to the rank of the words, but only related to the numbers of different words in a text (same assumption as bag-of-words (BOW) model [22]).

Conclusion: On above assumption, though each token contributes differently to text semantics, a larger difference (τ) in the numbers of mask tokens leads to a more accurate pseudo uncertainty relation.

Analysis: We token-mask an n-word text for twice. As a result, one augmented text has $e_0 + e_1$ remaining words, the other one has $e_0 + e_2$ remaining words after token-mask. Among the two augmented texts, e_0 words are the commonly remaining words in two texts after token-mask. The e_1 words and e_2 words are the two groups of uniquely remaining words for each augmented text, we assume $e_1 > e_2$. We set the semantic contributions of each word to a text for e_0 words as $\zeta_1, \zeta_2, ..., \zeta_{e_0}$, also set the semantic contributions of each word to a text for e_1 words and e_2 words as $\xi_1, \xi_2, ..., \xi_{e_1}$ and $\rho_1, \rho_2, ..., \rho_{e_2}$, respectively. Thus, based on BOW assumption, we have the semantics Ψ_1 and Ψ_2 of two augmented texts as,

$$\begin{aligned} \Psi_1 &= \sum_{i=1}^{e_0} \zeta_i + \sum_{j=1}^{e_1} \xi_j \\ \Psi_2 &= \sum_{i=1}^{e_0} \zeta_i + \sum_{j=1}^{e_2} \rho_j \end{aligned} \tag{12}$$

where each $\zeta_i \geq 0$, $\xi_j \geq 0$ and $\rho_j \geq 0$. Plus, each ζ_i, ξ_j and ρ_j in Eq. 12 is independent in BOW assumption. Due to the independence of each ζ_i, ξ_j, and ρ_j, the "each token contributes differently to text semantics" in our conclusion has been satisfied. Then, in the current situation (case 2 with no margin), due to $e_1 > e_2$, we assume $\Psi_1 > \Psi_2$, which means pseudo uncertainty scores of the first sample should be smaller than the pseudo uncertainty scores of the second sample. But the assumption might be wrong, because we have no idea whether $\sum_{j=1}^{e_1} \xi_j > \sum_{j=1}^{e_2} \rho_j$ or not. If $\sum_{j=1}^{e_1} \xi_j > \sum_{j=1}^{e_2} \rho_j$ is true, then our pseudo uncertainty relation is accurate. Thus, we define $P(\Psi_1 > \Psi_2)$ to represent a probability that our pseudo relation is accurate for a pair of augmented text, given $e_1 > e_2$.

In case 3, due to the additional margin τ, the difference in numbers of remaining words is enlarged. Thus, we token-mask less $\nu = \llcorner n \times \tau \lrcorner$ words for the first augmentation. As a result, the number of remaining words of the first augmentation is changed from $e_0 + e_1$ to $e_0 + e_1 + \nu$, but the number of remaining words of the second augmentation is still $e_0 + e_2$. Thus, the semantics $\Psi_1^{'}$ of the first augmented text with τ is,

$$\Psi_1^{'} = \sum_{i=1}^{e_0} \zeta_i + \sum_{j=1}^{e_1} \xi_j + \sum_{k=1}^{\nu} \xi_k^{'} \tag{13}$$

where each $\xi_k^{'} \geq 0$ is the semantic contributions of each word to a text for ν additionally remaining words. As a result, to compare the accuracy of pseudo uncertainty relation for a pair of augmented texts with/without τ, we have the below by plugging in Eq. 12 and Eq. 13,

$$\begin{aligned} &P(\Psi_1^{'} > \Psi_2) - P(\Psi_1 > \Psi_2) \\ &= (\Psi_1^{'} - \Psi_2) - (\Psi_1 - \Psi_2) \\ &= \Psi_1^{'} - \Psi_1 \geq 0 \end{aligned} \tag{14}$$

Thus, we show that the probability that the pseudo uncertainty relation with τ is true, is higher than that without τ. In other words, the accuracy of pseudo uncertainty relation with τ is higher.

A.2.2 Loss Modules. **Explain β in our revised cross-entropy loss L_{RC} of Eq. 7 in Sec. 4.4.** Given a prediction $\widehat{\mathbf{y}}_Q \in \mathbb{R}^k$ and its respective ground truth $\mathbf{y}_Q \in \mathbb{R}^k$, we analyze the the relation between $L_{CE}(\widehat{\mathbf{y}}_Q, \mathbf{y}_Q) + log(\beta)$ and 0, where $\beta \in [0.5, 1)$. We detail the L_{CE} as below,

$$L_{CE}(\widehat{\mathbf{y}}_Q, \mathbf{y}_Q) = -\sum_{i=1}^{k} \mathbf{y}_Q^i log(\widehat{\mathbf{y}}_Q^i) \tag{15}$$

where $\mathbf{y}_Q^i$ and $\widehat{\mathbf{y}}_Q^i$ are the i-th entry of $\mathbf{y}_Q$ and $\widehat{\mathbf{y}}_Q$ respectively. Since $\mathbf{y}_Q$ is a one-hot vector and we assume its $\mathbf{y}_Q^j = 1$ and the rest entries of $\mathbf{y}_Q$ are all 0, we have below,

$$\begin{aligned} L_{CE}(\widehat{\mathbf{y}}_Q, \mathbf{y}_Q) &= -\mathbf{y}_Q^j log(\widehat{\mathbf{y}}_Q^j) \\ &= -log(\widehat{\mathbf{y}}_Q^j) \end{aligned} \tag{16}$$

where $L_{CE} > 0$. This is because $\widehat{\mathbf{y}}_Q$ is input to a softmax function in the PyTorch implementation of cross-entropy loss [5], and $\widehat{\mathbf{y}}_Q^j \in (0, 1)$. However, we do not expect the $\widehat{\mathbf{y}}_Q^j$ is always close 1, as it is now not 100% confidence belonging to j-th class due to data augmentation. Thus, we add β to L_{CE} as below,

$$L_{CE}(\widehat{\mathbf{y}}_Q, \mathbf{y}_Q) + log(\beta) = -log(\widehat{\mathbf{y}}_Q^j) + log(\beta) = log(\frac{\beta}{\widehat{\mathbf{y}}_Q^j}) \tag{17}$$

Then, we substitute Eq. 17 into Eq. 7, we have below,

$$L_{RC} = max[log(\frac{\beta}{\widehat{\mathbf{y}}_{Q_1}^j}), 0] + max[log(\frac{\beta}{\widehat{\mathbf{y}}_{Q_2}^j}), 0] \tag{18}$$

where L_{RC} equals to constant 0 with no penality, if $\mathbf{y}_{Q_1}^j \geq \beta$ and $\mathbf{y}_{Q_2}^j \geq \beta$. Thus, they both have feasible solution $[\beta, 1)$ in L_{RC}.

[5] The PyTorch implementation of cross-entropy loss is "torch.nn.CrossEntropyLoss", which can be found from its official API.

Table 5: The comparison between baselines and our CLUR-b-3 in setting CNN as embeddings and Prototypical Network as classifiers on 20News with the 5-way 1-shot setting.

Methods	Uncertainty Ratio (F1 Score, Eliminated Ratio)↑					AUROC ↑	AUPR↑
	0%	10%	20%	30%	40%		
FTC-DS	27.12±3.58	35.75±3.43	43.48±3.29	51.64±3.09	58.96±2.95	55.75±5.96	37.11±6.91
DE	29.83±3.52	38.09±3.36	45.55±3.18	53.51±3.00	60.72±2.88	58.81±5.70	41.14±6.81
DE+Metric	31.09±3.04	39.22±2.89	46.54±2.76	54.35±2.56	61.35±2.41	58.76±4.74	42.17±5.05
MSD1	30.96±2.84	39.06±2.68	46.36±2.58	54.08±2.48	61.04±2.38	57.75±4.76	40.13±4.33
MSD2	30.36±3.53	38.44±3.34	45.71±3.17	53.53±2.99	60.60±2.77	57.72±5.26	40.54±5.85
SimSiam(CLUR-a-1)	30.39±3.42	38.52±3.28	45.81±3.14	53.55±2.97	60.66±2.76	57.58±5.32	40.62±5.90
CLUR-b-3	**31.77**±3.32	**40.16**±3.09	**47.54**±2.92	**55.37**±2.73	**62.47**±2.56	**59.20**±5.18	**43.89**±5.75

Table 6: Comparing baselines and CLUR-b-3 using FTC-DS on the Med-Domain dataset with the 5-way 1-shot setting.

Methods	Uncertainty Ratio (F1 Score, Eliminated Ratio)↑					AUROC ↑	AUPR↑
	0%	10%	20%	30%	40%		
FTC-DS	50.63±1.79	58.98±1.55	65.63±1.40	71.69±1.28	77.08±1.23	67.42±2.37	70.24±1.66
DE	56.01±1.83	63.13±1.67	69.36±1.53	75.17±1.44	80.36±1.32	70.94±2.54	75.53±1.43
DE+Metric	54.98±2.12	62.06±1.96	68.32±1.85	74.31±1.71	79.80±1.55	71.01±2.89	75.62±1.79
MSD1	55.93±1.99	62.88±1.82	69.04±1.70	74.85±1.60	80.02±1.44	70.10±2.71	74.39±1.65
MSD2	55.99±1.50	62.96±1.39	69.04±1.32	74.78±1.21	79.94±1.08	70.15±2.10	75.82±1.08
SimSiam(CLUR-a-1)	54.48±1.69	61.49±1.62	67.78±1.51	73.89±1.39	79.43±1.32	70.64±2.36	74.31±1.49
CLUR-b-3	**56.81**±1.69	**63.87**±1.51	**70.16**±1.42	**76.10**±1.32	**81.44**±1.21	**72.31**±2.36	**77.29**±1.31

Table 7: Parameter settings that we use to get our CLUR-b-3 results with fastText embeddings.

Datasets	5-way 1-shot				5-way 5-shot			
	γ	Φ_1	τ	Φ_2	γ	Φ_1	τ	Φ_2
20News	0.1	0.1	0.1	0.3	1	0.1	0.1	0.3
RCV1	1	0.1	0.1	0.3	1	0.1	0.05	0.25
Amazon	1	0.1	0.05	0.25	1	0.1	0.15	0.35
HuffPost	1	0.15	0.1	0.4	1	0.15	0.1	0.4

Below is our remaining losses, besides those in Sec. 4.4.

Loss module (a). In loss module (a), for an unequal uncertainty relation (case 2 or 3 in Sec. 4.2), our L_{CT_a} is,

$$\begin{aligned} L_{CT_a} &= max\{[H(\widehat{\mathbf{y}}_{Q_1}) - H(o(\widetilde{\mathbf{y}}_{Q_2}))] \times (\phi_2 - \phi_1), \\ &0\} + max\{[H(\widehat{\mathbf{y}}_{Q_2}) - H(o(\widetilde{\mathbf{y}}_{Q_1}))] \times (\phi_1 - \phi_2), 0\} \end{aligned} \tag{19}$$

The two items in Eq. 19 can be explained in a similar way to Eq. 10.

Loss module (c). It is designed to verify the effectiveness of detach in UEFTC. It has the same L_{RC} as Eq. 7. We only consider case 3 for it because we found case 3 achieved the best performance among three cases of uncertainty relations, when we used Loss module (b) for the experiments. Its L_{CT_c} in unequal uncertainty relation is,

$$L_{CT_c} = max\{[H(\widehat{\mathbf{y}}_{Q_1}) - H(\widehat{\mathbf{y}}_{Q_2})] \times (\phi_2 - \phi_1), 0\} \tag{20}$$

because there is no detach o, so there is no more difference between the two items in Eq. 10. The total loss is $L_{SUM_c} = L_{RC} + \gamma L_{CT_c}$.

Loss module (d). It is designed to verify the effectiveness of predictor in UEFTC by removing the predictors. Its L_{RC_d} is conducted on the projections $\widetilde{\mathbf{y}}_{Q_1}$. Its L_{CT_d} in unequal uncertainty relation is,

$$\begin{aligned} L_{CT_d} &= max\{[H(\widetilde{\mathbf{y}}_{Q_1}) - H(o(\widetilde{\mathbf{y}}_{Q_2}))] \times (\phi_2 - \phi_1), 0\} \\ &+ max\{[H(\widetilde{\mathbf{y}}_{Q_2}) - H(o(\widetilde{\mathbf{y}}_{Q_1}))] \times (\phi_1 - \phi_2), 0\} \end{aligned} \tag{21}$$

The total loss is $L_{SUM_d} = L_{RC_d} + \gamma L_{CT_d}$.

A.3 Experiments

A.3.1 Implementation Details. We use fastText [33] as the word embedding by default. For all experiments, we set $\beta = 0.75$. We list the parameter settings on Tab. 7, which are parameters used to get our reported CLUR-b-3 results. For the 5-way 1-shot using BERT embedding on 20News, we set $\gamma = 0.1$, $\Phi_1 = 0.1$, $\tau = 0.1$, and $\Phi_2 = 0.3$. The ablation studies also use the respective parameters.

A.3.2 Generalization On Other Few-Shot Model. **Our CLUR-b-3 is experimentally effective on CNN embedding and Prototypical Network classifier.** We conducted experiments using CNN embeddings [68] and Prototypical Network [2] classifiers for the UEFTC, which are common used in few-shot learning. We compared our CLUR-b-3 model to the baselines in the 5-way 1-shot setting on the 20News dataset. The results of our experiments are presented in Tab. 5. Our results indicate that our CLUR-b-3 model surpasses all the baselines, such as achieving over 3.35 points AURP compared to MSD2 in Tab. 5. These results suggest that our CLUR-b-3 model is effective not only for FTC-DS [1] but also for CNN embeddings and Prototypical Network classifiers. As a result, CLUR exhibits potential for generalization to other few-shot models.

A.3.3 Experiment On A Med-Domain Dataset. Besides the four common-used datasets, we also explore the effectiveness of CLUR on public high-risk datasets like healthcare. We choose Med-Domain dataset [10], which has medical transcription and labels. We use its subset of 1294 samples among 29 classes for our experiments, where the subset is released in our data split. We compare the performance between baselines and our CLUR-b-3 based on FTC-DS, shown as Tab. 6. We conclude that our CLUR still works well in the high-risk domain, say, the healthcare domain. This is because our CLUR surpasses all baselines, such as improving 2.16 points AUROC over MSD2 in Tab. 6.