CUE: An Uncertainty Interpretation Framework for Text Classifiers Built on Pre-Trained Language Models

Jiazheng Li[1] **Zhaoyue Sun**[2] **Bin Liang**[3] **Lin Gui**[1] **Yulan He**[1,2,4]

[1]Department of Informatics, King's College London, UK
[2]Department of Computer Science, University of Warwick, UK
[3]Joint Lab of HITSZ-CMS, Harbin Institute of Technology, Shenzhen, China
[4]The Alan Turing Institute, UK

Abstract

Text classifiers built on Pre-trained Language Models (PLMs) have achieved remarkable progress in various tasks including sentiment analysis, natural language inference, and question-answering. However, the occurrence of uncertain predictions by these classifiers poses a challenge to their reliability when deployed in practical applications. Much effort has been devoted to designing various probes in order to understand what PLMs capture. But few studies have delved into factors influencing PLM-based classifiers' predictive uncertainty. In this paper, we propose a novel framework, called `CUE`, which aims to interpret uncertainties inherent in the predictions of PLM-based models. In particular, we first map PLM-encoded representations to a latent space via a variational auto-encoder. We then generate text representations by perturbing the latent space which causes fluctuation in predictive uncertainty. By comparing the difference in predictive uncertainty between the perturbed and the original text representations, we are able to identify the latent dimensions responsible for uncertainty and subsequently trace back to the input features that contribute to such uncertainty. Our extensive experiments on four benchmark datasets encompassing linguistic acceptability classification, emotion classification, and natural language inference show the feasibility of our proposed framework. Our source code is available at `https://github.com/lijiazheng99/CUE`.

1 INTRODUCTION

Text classifiers built on Pre-trained Language Models (PLMs) have made remarkable progress on various Natural Language Processing (NLP) tasks [Devlin et al., 2019, Liu et al., 2020b, Lan et al., 2020, Sanh et al., 2019]. However, their deployment in practical applications still faces significant challenges. Of particular concern, these models tend to make over-confident predictions in uncertain cases [Guo et al., 2017, He et al., 2020, Malinin and Gales, 2018]. Since PLMs have been widely used in various applications, such issues cause concerns about model trustworthiness and transparency, which becomes a barrier to deploying PLMs in sensitive domains such as medicine and finance.

Predictive uncertainty is generally believed to include two aspects - *aleatoric uncertainty* and *epistemic uncertainty*, where the aleatoric uncertainty measures the data uncertainty due to inherent random effects and is irreducible, while the epistemic uncertainty measures the uncertainty caused by the lack of knowledge from data and is reducible [Malinin and Gales, 2018]. Numerous approaches have been proposed to estimate the predictive uncertainty of deep neural models, such as Deep Ensemble models [Lakshminarayanan et al., 2017], Bayesian Neural Networks (BNN) [Blundell et al., 2015] and Monte-Carlo (MC) Dropout [Gal and Ghahramani, 2016]. Similar idea has been applied to PLMs in recent years, to study the uncertainty of text classifiers. Particularly, quantifying uncertainty in PLM-based classifiers can be done by incorporating weight uncertainty into the PLM architecture. However, uncertainty can only be induced to a certain number of layers (e.g., the last layer of the PLM feature extractor and/or the classification layer) due to a large number of PLM layers and parameters. Alternatively, one could use deep ensembles by aggregating classification results generated from multiple PLM classifiers trained with different initialisation [Lakshminarayanan et al., 2017], or apply MC dropout in the inference stage to estimate the uncertainty of PLMs [Vazhentsev et al., 2022].

Previous studies [Vulić et al., 2020, Clark et al., 2019, Yang et al., 2021] have also been devoted to designing various probes in order to understand what PLMs capture. Nevertheless, they largely ignore the interpretation of the source of the uncertainty, i.e., identifying the input features which cause classification uncertainty, which can be crucial for

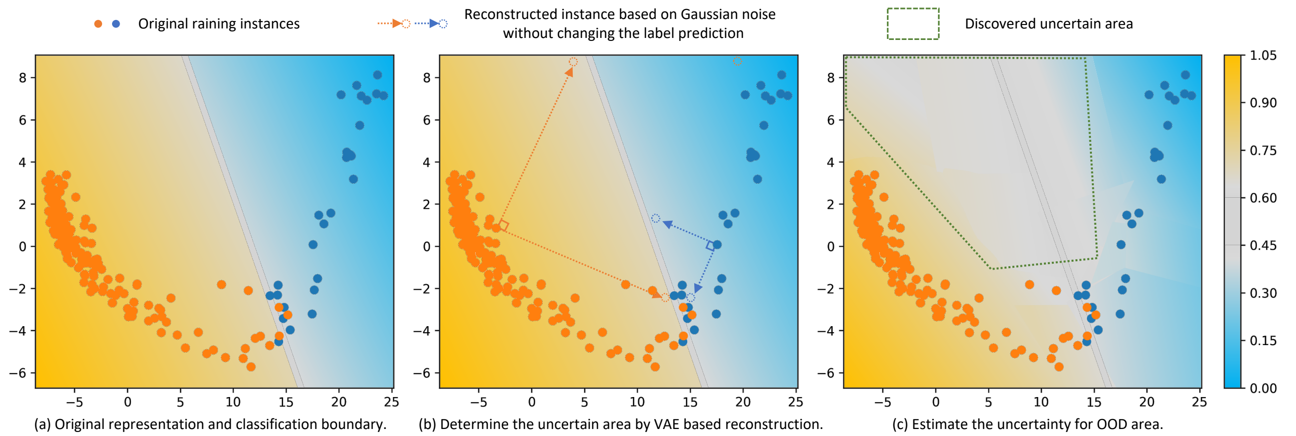


Figure 1: The illustration of proposed framework: **(a)** The original representations learned from a Pre-trained Language Model (PLM) and the decision boundary separating two classes; **(b)** We can freeze the PLM-classifier parameters and perturb the PLM-encoded representations in the latent space to increase the aleatoric (data) uncertainty along some isotropic directions while preserving the predictive labels. This would help determine the uncertain areas; **(c)** Using the representations reconstructed from the perturbed latent space, the uncertainty for out-of-distribution (OOD) areas can be estimated.

understanding the model and taking appropriate mitigating strategies. In text classification, recent research tried to identify word tokens that lead to uncertainty via perturbations on input sequences [Shen et al., 2020, Kim et al., 2020]. However, due to the discreteness of textual data, token replacement or removal would require a large search space on the input sequence and incur expensive computational costs.

In this paper, we aim to interpret the predictive uncertainty on PLM text classifiers by identifying the input tokens that cause the uncertainty. We propose a novel PLM **C**lassifier **U**ncertainty **E**xplanation (`CUE`) framework built on Variational Auto-encoder (VAE) [Kingma and Welling, 2014, Card et al., 2018] that generates perturbations on latent text representations to induce uncertain predictions. As shown in Figure 1, we can perturb the PLM-encoded representations in the latent space to increase the aleatoric uncertainty (data uncertainty) along some isotropic directions while preserving the predictive labels. As will be shown in §3.2, this is equivalent to decreasing the predictive epistemic uncertainty. By examining the difference between the original and the perturbed text representations, a subset of input features (i.e., word tokens) can be identified as the interpretation of the original model's predictive uncertainty. We compared our framework with existing approaches addressing the predictive uncertainty problem on three classification tasks across four benchmark datasets. Extensive experimental results show that our proposed method can identify the source of epistemic uncertainty and calibrate text representations from four commonly used PLMs.

In summary, our contributions are: **(1)** We propose a novel framework `CUE` to induce perturbations on PLM-encoded representations for uncertainty interpretation of the PLM-based text classifiers. **(2)** We propose an uncertainty feature identification algorithm to identify token-level features which lead to model predictive uncertainty. **(3)** We validate the effectiveness of our proposed framework by conducting extensive experiments using various classifiers built on four commonly-used PLMs on three different tasks and four datasets with class numbers ranging from 2 to 27. The results show that our proposed framework achieves lower expected calibration errors compared to existing approaches such as label smoothing, MC dropout, and BNN. To the best of our knowledge, our framework is the first to study the token interpretation of PLM-based classifiers' predictive uncertainty from the representation space, without editing the semantic meaning of the original input text.

2 RELATED WORK

Our work is related to two lines of research, interpretation of PLMs and uncertainty estimation in ML.

Interpretation of PLMs Transformer-based language models have achieved impressive performance across various NLP tasks [Devlin et al., 2019, Liu et al., 2020b]. However, the complex structure of these models has raised concerns about model transparency and reliability. Thus, there has been growing interest in developing methods to interpret PLMs. For example, Clark et al. [2019] proposed an attention-based visualisation method to interpret the model parameters by probing the feature space to determine the potential influence of the model output. Brunner et al. [2020] studied the identifiability of attention weights in the BERT model and found that the distribution of self-attentions cannot be directly used as an interpretation. There has also been work focusing on interpreting the representations from

PLMs [Zhou and Srikumar, 2021] and attention weights [Sun and Marasović, 2021, Mareček and Rosa, 2019].

Uncertainty Estimation As the interpretation of PLMs cannot provide prediction confidence directly, much effort has been devoted to developing approaches for uncertainty estimation of neural models. A straightforward approach to uncertainty estimation is using deep ensemble models [Lakshminarayanan et al., 2017]. Various Bayesian inference methods have also been developed to prevent overfitting by attaching distributions to parameters in standard networks and estimating parameters via posterior inference [Blundell et al., 2015]. Alternatively, uncertainty estimation can be performed using MC dropout [Gal and Ghahramani, 2016], which performs multiple stochastic forwards passes with dropout in a network during the inference stage to produce an ensemble of predictions. Other approaches to uncertainty estimation include prior networks [Malinin and Gales, 2018]. Taking advantage of the development of transformer [Vaswani et al., 2017], there has been increasing interest in investigating classification uncertainty of language models [Xiao and Wang, 2019, Desai and Durrett, 2020]. Various methods have been developed, partly inspired by the research in computer vision, from uncertainty quantification via input marginalization [Shen et al., 2020, Kim et al., 2020] to MC dropout and Bayesian inference methods such as SNGP [Shelmanov et al., 2021, Vazhentsev et al., 2022, Liu et al., 2020a]. Nevertheless, the aforementioned approaches cannot identify the cause of the uncertainty.

To overcome the limitation of existing methods, we propose an uncertainty analysis framework `CUE` built on VAE [Kingma and Welling, 2014, Card et al., 2018, Qiu et al., 2020], in which noise can be generated by perturbing the latent representation space. This allows us to disentangle the source of uncertainty via text representation dimensions and study PLM-based classifiers' predictive uncertainty at both the sequence- and the token-level.

3 BACKGROUND

3.1 PROBLEM SETUP

We are given a labelled text classification dataset, where X is input text and Y is the label set. $\forall\{x_i, y_i\} \in X \times Y$, where $(x_i, y_i), i = 1, 2, ..., N$, is an i.i.d. realisation of the random variables, $P(Y|X) \sim \mathbb{D}$, where $\mathbb{D}$ is the unknown ground truth conditional distribution of class labels. To train a text classifier built on a PLM, we need to find an optimal feature extraction function f and a classification layer g with trainable parameter $\boldsymbol{\vartheta}$ and $\boldsymbol{\eta}$, respectively: $x_i \xrightarrow{f_{\boldsymbol{\vartheta}}(\cdot)} \boldsymbol{e}_i \xrightarrow{g_{\boldsymbol{\eta}}(\cdot)} \hat{y}_i$, which first encodes text into a representation $\boldsymbol{e}_i$ and then outputs a probability distribution over the label set with the predicted output close to the desired true label y_i. In this work, we take one step further to analyse the potential uncertainty in the two stages of the learning process: 1) in $\boldsymbol{\eta}$: $\boldsymbol{e}_i \xrightarrow{g_{\boldsymbol{\eta}}(\cdot)} \hat{y}_i$, which dimension(s) in $\boldsymbol{e}_i$ is the source of uncertainty in prediction; and 2) in $\boldsymbol{\vartheta}$: $x_i \xrightarrow{f_{\boldsymbol{\vartheta}}(\cdot)} \boldsymbol{e}_i$, which input tokens cause the uncertainty. Before we detail our proposed uncertainty estimation approach, we give the formal definition of uncertainty first.

3.2 UNCERTAINTY ESTIMATION

According to established definitions found in prior literature, uncertainty can be defined based on the probability of predictive error [Sullivan, 2015], the mean squared error (MSE) [Cervera et al., 2021], or the conditional entropy [Malinin and Gales, 2021]. We adopt the MSE-based definition as a representative measure of uncertainty, which is chosen without compromising the generality of our approach.

Definition 3.1. $\forall P(y|x) \in \mathbb{D}$, the predictive epistemic uncertainty can be defined by $\mathbb{E}\big[\big(\mathbb{E}[y] - \hat{y}\big)^2\big]$.

Here, $\hat{y}$ is the class label for input x predicted by the trained classifier. $\mathbb{E}[y]$ is the expectation of the ground truth label distribution, which is however unknown to the learner, making it impossible to calculate the epistemic uncertainty based on predictive variance directly. Therefore, we propose to estimate the uncertainty by decomposing the variance based on the observed training data, $\{x_i, y_i\}_{i=1}^N$, which yields:

$$\mathbb{E}[(y_i - \hat{y}_i)^2] = \underbrace{\mathbb{E}[(y_i - \mathbb{E}[y])^2]}_{\text{aleatoric uncertainty}} + \underbrace{\mathbb{E}[(\mathbb{E}[y] - \hat{y}_i)^2]}_{\text{epistemic uncertainty}} \tag{1}$$

Since the first term, $\mathbb{E}[(y_i - \mathbb{E}[y])^2]$, contains the observed y_i, it can be defined as the aleatoric uncertainty. The detailed derivation of Eq. (1) can be found in our Supplementary Material Section 1.1. Similar to the setup in [Heiss et al., 2023], if we assume the conditional distribution of class labels follows a Gaussian distribution:

Assumption 3.2. $\forall P(y_i|x_i) \in \mathbb{D}$, the true label distribution for a give data x_i follows a Gaussian noise based generating process: $P(y_i|x_i) = \boldsymbol{\mu}_y + \boldsymbol{\varepsilon}_{y_i}$, where $\boldsymbol{\mu}_y = \mathbb{E}[y]$ and the noise $\boldsymbol{\varepsilon}_{y_i}$ follows a Gaussian distribution of $\mathcal{N}(0, \boldsymbol{\sigma}_y^2)$ and $\boldsymbol{\sigma}_y^2 = \mathbb{E}[(\mathbb{E}[y] - y_i)^2]$.

Then, the epistemic uncertainty can be written as:

$$\mathbb{E}\big[(\mathbb{E}[y] - \hat{y}_i)^2\big] = \mathbb{E}[(y_i - \hat{y}_i)^2] - \boldsymbol{\sigma}_y^2. \tag{2}$$

Here, the term $\mathbb{E}[(y_i - \hat{y}_i)^2]$ in the Eq. (2) is the empirical MSE on the training data which can be optimised in the training process. The term $\boldsymbol{\sigma}_y^2$ is based on the true label distribution which is unseen to the learner. We need to clarify that $P(y_i|x_i)$ can be larger than 1 under the assumption. Therefore, there is necessary to stack a normalisation layer before the prediction to guarantee the sum of predictive probabilities for different class labels is 1.

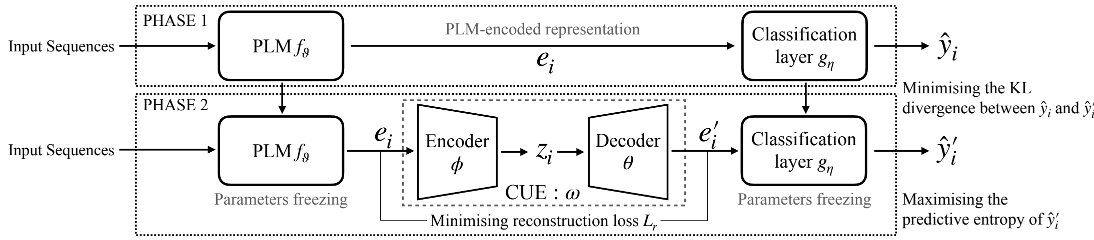


Figure 2: CUE requires two phases of training: **Phase 1**: Fine-tune a classifier built on a PLM on a target dataset; **Phase 2**: Freeze the PLM and classification layer parameters and plug-in the CUE module to train the model that perturbs the latent representation z_i to generate e_i' which leads to uncertainty fluctuation.

We assume that the empirical MSE has been minimised by the trained PLM-based classifier with parameters $\boldsymbol{\vartheta}$ and $\boldsymbol{\eta}$. To minimise the epistemic uncertainty given by Eq. (2) for a given x_i and its corresponding representation $\boldsymbol{e}_i$, we have to increase $\boldsymbol{\sigma}_y^2$, which however cannot be calculated directly. We propose to use a VAE-based generative model parameterised by $\boldsymbol{\omega}$ to reconstruct $\boldsymbol{e}_i$ by adding Gaussian noise while preserving the predictive label, resulting in $\boldsymbol{e}_i'$. The reconstructed representation should be similar to the original input representation, $\boldsymbol{e}_i' \approx \boldsymbol{e}_i$, and the predictive class label distribution from $\boldsymbol{e}_i'$, $\hat{y}_i' = g_{\boldsymbol{\eta}}(\boldsymbol{e}_i')$, should be close to $\hat{y}_i = g_{\boldsymbol{\eta}}(\boldsymbol{e}_i)$, $\hat{y}_i' \approx \hat{y}_i$. This allows us to manipulate the latent code of the VAE to increase the variance of the Gaussian noise, which leads to the resulting label distribution closer to a uniform distribution in the out-of-distribution (OOD) area, thus achieving a lower epistemic uncertainty. Accordingly, we define the learning objective function as:

Learning objective: $\forall P(y_i|x_i) \in \mathbb{D}$ under the Assumption 3.2, the learning objective is to:

$$\min_{\boldsymbol{\omega}} \mathbb{E}[(y_i - \hat{y}_i)^2] - \mathcal{H}_{e'}(\hat{y}), \quad s.t. \quad \boldsymbol{e}_i \approx \boldsymbol{e}_i', \quad \hat{y}_i \approx \hat{y}_i', \tag{3}$$

where $\boldsymbol{\omega}$ denotes the parameters of VAE, $\boldsymbol{e}_i = f_{\boldsymbol{\vartheta}}(x_i)$, $\boldsymbol{e}_i' = f_{\boldsymbol{\omega}}(\boldsymbol{e}_i)$, $\hat{y}_i = g_{\boldsymbol{\eta}}(\boldsymbol{e}_i)$, $\hat{y}_i' = g_{\boldsymbol{\eta}}(\boldsymbol{e}_i')$, $\mathcal{H}_{e'}(\hat{y})$ is the estimated entropy by the predictive label distribution from the reconstructed $\boldsymbol{e}_i'$, which approximates the variance of the true label distribution. The above learning objective can be formulated using the method of Lagrange multipliers:

$$\begin{aligned}\mathcal{L}(x_i, y_i) = &- \mathbb{E}[\mathcal{H}_{e'}(\hat{y}_i)] + \lambda_1 \mathbb{E}[(f_{\boldsymbol{\vartheta}}(x_i) - f_{\boldsymbol{\omega}}(f_{\boldsymbol{\vartheta}}(x_i)))^2] \\ &+ \lambda_2 \mathrm{KL}(\hat{y}_i' || \hat{y}_i)\end{aligned} \tag{4}$$

Therefore, by optimising Eq. (4), we can obtain an alternative representation of $\boldsymbol{e}_i'$ with the predictive distribution of $\hat{y}_i'$ using the parameters $\boldsymbol{\omega}$, where the lower bound of the epistemic uncertainty can be obtained by increasing the aleatoric uncertainty defined by $\mathcal{H}_{e'}(\hat{y}_i)$. In the next section, we show how each term in Eq. (4) can be defined in our VAE-based uncertainty interpretation framework CUE.

4 UNCERTAINTY INTERPRETATION

In this paper, we are interested in interpreting model uncertainty, that is, what input features lead to the predictive uncertainty. To this end, we propose a VAE-based uncertainty interpretation framework CUE, as shown in Figure 2. Rather than directly perturbing the input features, perturbations can be done in the latent space in CUE to generate the modified input representation such that it still resides on the original data manifold while the model's predictive epistemic uncertainty on the modified input is reduced. By examining the difference between the original and the perturbed text representations, a subset of input features (i.e., word tokens) can be identified as the interpretation of the original model's predictive uncertainty.

We will first present how to generate perturbations on latent space in order to cause the prediction uncertainty change (§4.1). We will then describe how to identify input features that lead to original prediction uncertainty to facilitate the interpretation of model predictive uncertainty (§4.2).

4.1 LATENT SPACE PERTURBATION FOR EPISTEMIC UNCERTAINTY REDUCTION

Once a classifier built on a PLM is fine-tuned on a target dataset, we freeze the parameters of the PLM and the classification layer and then insert the CUE between the PLM last layer and the task-specific classification layer. The PLM-encoded representation $\boldsymbol{e}_i$ is mapped to a latent vector, denoted by $\boldsymbol{z}_i$, via CUE which consists of two networks.

The *encoder* network $\boldsymbol{\phi}$, learns the distribution of a lower dimensional latent variable $\boldsymbol{z}_i$ given the PLM-encoded representation by a random Gaussian noise ϵ: $\boldsymbol{z}_i = \boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{e}_i) + \epsilon \cdot \boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{e}_i)$, i.e., $\boldsymbol{z}_i \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\phi}}, \boldsymbol{\sigma}_{\boldsymbol{\phi}}^2)$.

The *decoder* network $\boldsymbol{\theta}$, reconstructs the text representation given the latent variable $\boldsymbol{z}_i$, defined as $\boldsymbol{e}_i' = p_{\boldsymbol{\theta}}(\boldsymbol{z}_i)$. Although $p_{\boldsymbol{\theta}}(\boldsymbol{z}_i)$ can be any decoding network, our implementation utilises a linear mapping $\boldsymbol{W}_{\boldsymbol{\theta}}$ without a bias term. The benefit is that $\boldsymbol{W}_{\boldsymbol{\theta}}$ can be treated as a set of learnable vectors and the reconstructed text representation $\boldsymbol{e}_i'$ can be written as a linear combination of the decoded output generated from each of the latent dimensions of $\boldsymbol{z}_i$. As will be discussed in §4.2, such a decomposition form of decoding as illustrated in Eq. (10) allows the identification of latent dimensions of $\boldsymbol{z}_i$ which causes predictive uncertainty.

The VAE parameters are denoted as $\boldsymbol{\omega} = \{\boldsymbol{\phi}, \boldsymbol{\theta}\}$. The classifier's prediction on the reconstructed representation $\boldsymbol{e}'_i$ is denoted as $\hat{y}'_i = g_{\boldsymbol{\eta}}(\boldsymbol{e}'_i)$. Here, we choose to use the Softmax based prediction layer to normalise the predictive probability, but the representation $\boldsymbol{e}'_i$ before the normalisation should follow the Gaussian distribution since it is captured by a linear combination of Gaussians. Besides, the latent representation $\boldsymbol{z}_i$ can be perturbed which leads to uncertain predictions bounded by a uniform distribution probability, $\log K$ (K is the label set size)[1]. For the training of the CUE model, we define various loss terms in Eq. (4) below:

Minimum change on both the perturbed representation and the model prediction. The reconstructed $\boldsymbol{e}'_i$ should be similar to the original $\boldsymbol{e}_i$.

$$\mathcal{L}_r = \left\|\boldsymbol{e}'_i - \boldsymbol{e}_i\right\|^2 \tag{5}$$

The prediction, $p(\hat{y}'_i|\boldsymbol{e}'_i)$, based on the reconstructed e'_i, should be close to the origin prediction $p(\hat{y}_i|\boldsymbol{e}_i)$.

$$\mathrm{KL}(\hat{y}'_i||\hat{y}_i) = \sum_{k=1}^{K} p(\hat{y}'_{i_k}|\boldsymbol{e}'_i) \log \frac{p(\hat{y}'_{i_k}|\boldsymbol{e}'_i)}{p(\hat{y}_{i_k}|\boldsymbol{e}_i)}, \tag{6}$$

where K denotes the size of the class label set.

Predictive Entropy Incrememt. We need to increase the predictive entropy $\mathcal{H}_{e'}(\hat{y}'_i)$ calculated based on the reconstructed input representation $\boldsymbol{e}'_i$, which approximates the variance of the true label distribution, $\boldsymbol{\sigma}^2_y$, in order to decrease the model epistemic uncertainty defined in Eq. (2).

$$\mathcal{H}_{e'}(\hat{y}'_i) = -\sum_{k=1}^{K} p(\hat{y}'_{i_k}|\boldsymbol{e}'_i) \log p(\hat{y}'_{i_k}|\boldsymbol{e}'_i) \tag{7}$$

In addition, we incorporate an orthogonality constraint within the decoder to encourage independence among dimensions of the latent variable:

$$\mathcal{L}_o = \|\mathbf{I} - \boldsymbol{W_\theta} \times \boldsymbol{W_\theta}^{\mathsf{T}}\| \tag{8}$$

where $\mathbf{I}$ is an identity matrix, W_θ is the weights in the decoder. The final objective function is then defined as:

$$\mathcal{L} = \gamma_1 \mathcal{L}_r + \gamma_2 \mathrm{KL}(\hat{y}'_i||\hat{y}_i) - \gamma_3 \mathcal{H}_{e'}(\hat{y}'_i) + \gamma_4 \mathcal{L}_o \tag{9}$$

where the γ coefficients are used to balance various loss terms. Minimising the loss function defined in Eq. (9) is equivalent to introducing perturbation in the latent space so as to increase the predictive entropy. We can use the reconstruction error $||\boldsymbol{e}'_i - \boldsymbol{e}_i||^2$ to represent the perturbed noise that leads to predictive uncertainty difference $\Delta\mathcal{H}$. As will be shown in Supplementary Material Section 1.2, $\Delta\mathcal{H}$ is proportional to the reconstruction error $||\boldsymbol{e}'_i - \boldsymbol{e}_i||^2$. As such, the reconstruction error can be used to interpret the predictive uncertainty. By retracing alterations made in the input feature space, we can effectively identify features which cause the uncertainty. To the best of our knowledge, we are the first to apply perturbations in the latent representation space to interpret the predictive uncertainty associated with PLM-based classifiers.

[1]The proof is shown in Supplementary Material Section 1.2.

4.2 INPUT FEATURE IDENTIFICATION FOR UNCERTAINTY INTERPRETATION

In this subsection, we discuss how to quantify the prediction uncertainty that caused by the input features based on the latent space perturbation in §4.1. The discussion is built on an inner product space defined by our noise generation methods. During the inference stage, we identify the possible feature that caused predictive uncertainty by our proposed Uncertain Feature Identification (UFI) algorithm[2]. For a given input, we can retrieve three different representations from the CUE framework, the original PLM-encoded representation $\boldsymbol{e}_i$, the reconstructed representation $\boldsymbol{e}'_i$, and the difference between two representation, $\Delta e_i = \boldsymbol{e}'_i - \boldsymbol{e}_i$. The reconstructed representation $\boldsymbol{e}'_i$ can be rewritten as the weighted sum of each latent dimension from $\boldsymbol{z}_i$, where the weight is given by the decoder:

$$\boldsymbol{e}'_i = \Sigma_{d=1}^{\dim} q_\phi(\boldsymbol{z}_{i\,d}|\boldsymbol{e}_i) \cdot \boldsymbol{r}_{z_{i\,d}}, \quad \boldsymbol{r}_{z_{i\,d}} = \mu_\theta(\boldsymbol{z}_{i\,d}), \tag{10}$$

where $\dim$ is the size of the latent space and $\boldsymbol{r}_{z_{i\,d}}$ denotes the representation generated via the d-th dimension's code corresponding to the latent vector $\boldsymbol{z}_i$ from the decoder. As mentioned in §4.1, $\Delta\mathcal{H}$ is proportional to the reconstruction error $||\boldsymbol{e}'_i - \boldsymbol{e}_i||^2$. We thus use the norm (calculated as the inner product) of the reconstruction error, $||\Delta\boldsymbol{e}_i||^2 = \langle\Delta\boldsymbol{e}_i, \Delta\boldsymbol{e}_i\rangle$, to measure the entropy change as:

$$\begin{aligned}\langle\Delta\boldsymbol{e}_i, \Delta\boldsymbol{e}_i\rangle &= \langle\Sigma_{d=1}^{\dim} q_\phi(\boldsymbol{z}_{i\,d}|\boldsymbol{e}_i) \cdot \boldsymbol{r}_{z_{i\,d}} - \boldsymbol{e}_i, \Delta\boldsymbol{e}_i\rangle \\ &= \langle\Sigma_{d=1}^{\dim} q_\phi(\boldsymbol{z}_{i\,d}|\boldsymbol{e}_i) \cdot \boldsymbol{r}_{z_{i\,d}}, \Delta\boldsymbol{e}_i\rangle - \langle\boldsymbol{e}_i, \Delta\boldsymbol{e}_i\rangle,\end{aligned} \tag{11}$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product. In the first line of Eq. (11), we substitute the first $\Delta\boldsymbol{e}_i$ with $\boldsymbol{e}'_i - \boldsymbol{e}_i$, and further substitute $\boldsymbol{e}'_i$ with Eq. (10). When determining the relative importance of each latent dimension with respect to the predictive entropy change, $\langle\boldsymbol{e}_i, \Delta\boldsymbol{e}_i\rangle$ can be ignored as it is the same for all latent dimensions. Therefore, the inner product of $\langle\boldsymbol{r}_{z_{i\,d}}, \Delta\boldsymbol{e}_i\rangle$, which dominates the norm value of $||\Delta\boldsymbol{e}_i||$ in the d-th dimension can be used to measure predictive uncertainty caused by each dimension from the latent space $\boldsymbol{r}_{z_i}$, and thus determine each dimension's importance.

On the other hand, the input text representation $\boldsymbol{e}_i$ output by the PLM at layer-L can be written as a Softmax-based weighted sum of each token's representation from the previous layer $L-1$ by[3]:

$$\begin{aligned}\boldsymbol{e}_i^L &= \Sigma_{j=1}^{n} \mathrm{Softmax}(\langle\boldsymbol{e}_i, \boldsymbol{e}_{ij}\rangle) \cdot \boldsymbol{e}_{ij} \\ &\propto \Sigma_{j=1}^{n} \exp(\langle\boldsymbol{e}_i, \boldsymbol{e}_{ij}\rangle) \cdot \boldsymbol{e}_{ij},\end{aligned} \tag{12}$$

[2]We provide the UFI algorithm implementation in Supplementary Material Section 2.

[3]Note that all representations in the RHS are from Layer $L-1$. We drop the superscript $L-1$ to simplify the notations.

Model	CoLA				GoEmotions			
	Acc	F1	$\mathcal{H}$	ECE↓	Acc	F1	$\mathcal{H}$	ECE↓
ALBERT (11M)	0.7923±0.0192	0.8624±0.0106	0.4650±0.0695	0.0834±0.0179	0.6193±0.0051	0.4545±0.0106	0.3574±0.0042	0.0446±0.0080
ALBERT Label Smoothing	0.7699±0.0452	0.8508±0.0195	0.5712±0.1834	0.0625±0.0345	0.6219±0.0025	0.4579±0.0133	0.3651±0.0163	0.0353±0.0146
ALBERT MC Dropout	0.7893±0.0109	0.8583±0.0065	0.4575±0.0589	0.0854±0.0223	0.6152±0.0055	0.4448±0.0144	0.3697±0.0188	**0.0345**±0.0076
ALBERT w/ BNN	0.7973±0.0011	0.8647±0.0006	0.4180±0.0015	0.0936±0.0016	0.6187±0.0009	0.4396±0.0015	0.3170±0.0003	0.0864±0.0011
ALBERT w/ `CUE` (Ours)	0.8038±0.0005	0.8668±0.0004	0.5771±0.0004	**0.0444**±0.0031	0.6176±0.0021	0.4567±0.0046	0.3814±0.0294	0.0395±0.0098
DistilBERT (66M)	0.7634±0.0032	0.8479±0.0019	0.5412±0.0151	0.0842±0.0060	0.6231±0.0018	0.4637±0.0047	0.3312±0.0024	0.0566±0.0039
DistilBERT Label Smoothing	0.7632±0.0033	0.8477±0.0018	0.5620±0.0154	0.0765±0.0063	0.6233±0.0021	0.4643±0.0042	0.3412±0.0033	0.0520±0.0035
DistilBERT MC Dropout	0.7787±0.0241	0.8559±0.0125	0.4773±0.0864	0.0897±0.0110	0.6224±0.0023	0.4670±0.0054	0.3246±0.0033	0.0623±0.0025
DistilBERT w/ BNN	0.7659±0.0016	0.8491±0.0010	0.5133±0.0002	0.0966±0.0021	0.6237±0.0009	0.4550±0.0016	0.3080±0.0002	0.0802±0.0005
DistilBERT w/ `CUE` (Ours)	0.7831±0.0012	0.8540±0.0010	0.8362±0.0008	**0.0738**±0.0029	0.6253±0.0017	0.4517±0.0022	0.4457±0.0004	**0.0208**±0.0031
BERT (110M)	0.8000±0.0072	0.8696±0.0043	0.4026±0.0317	0.0995±0.0057	0.6266±0.0032	0.4829±0.0075	0.3342±0.0049	0.0537±0.0047
BERT Label Smoothing	0.8036±0.0085	0.8717±0.0044	0.4099±0.0843	0.0967±0.0185	0.6268±0.0040	0.4819±0.0086	0.3444±0.0046	0.0492±0.0020
BERT MC Dropout	0.8008±0.0054	0.8703±0.0037	0.4023±0.0321	0.0987±0.0100	0.6266±0.0026	0.4889±0.0085	0.3337±0.0056	0.0548±0.0053
BERT w/ BNN	0.6104±0.1689	0.6545±0.3509	0.8837±0.0732	0.1095±0.1391	0.6296±0.0008	0.4855±0.0008	0.3102±0.0001	0.0775±0.0014
BERT w/ `CUE` (Ours)	0.8123±0.0012	0.8762±0.0007	0.4991±0.0032	**0.0677**±0.0056	0.6282±0.0029	0.4712±0.0087	0.4433±0.0159	**0.0326**±0.0013
RoBERTa (125M)	0.8050±0.0142	0.8721±0.0072	0.3310±0.0472	0.1100±0.0190	0.6226±0.0051	0.4877±0.0095	0.3310±0.0472	0.0602±0.0073
RoBERTa Label Smoothing	0.8165±0.0128	0.8788±0.0058	0.3091±0.0415	0.1079±0.0120	0.6215±0.0021	0.4866±0.0098	0.3091±0.0415	0.0554±0.0084
RoBERTa MC Dropout	0.8056±0.0056	0.8724±0.0023	0.3340±0.0452	0.1074±0.0165	0.6217±0.0034	0.4907±0.0082	0.3340±0.0452	0.0604±0.0060
RoBERTa w/ BNN	0.7992±0.0022	0.8699±0.0014	0.3069±0.0010	0.1228±0.0021	0.6227±0.0002	0.4686±0.0056	0.3069±0.0010	0.0881±0.0002
RoBERTa w/ `CUE` (Ours)	0.8075±0.0087	0.8744±0.0045	0.6077±0.0595	**0.0465**±0.0075	0.6255±0.0005	0.4540±0.0013	0.6077±0.0595	**0.0316**±0.0024
	Emotion				MultiNLI			
Model	Acc	F1	$\mathcal{H}$	ECE↓	Acc	F1	$\mathcal{H}$	ECE↓
ALBERT (11M)	0.9284±0.0037	0.8862±0.0031	0.8862±0.0031	0.0348±0.0059	0.8362±0.0018	0.8358±0.0018	0.8358±0.0018	0.0465±0.0036
ALBERT Label Smoothing	0.9310±0.0023	0.8897±0.0043	0.8897±0.0043	**0.0231**±0.0018	0.8327±0.0020	0.8317±0.0020	0.8317±0.0020	0.0364±0.0034
ALBERT MC Dropout	0.9331±0.0033	0.8927±0.0036	0.8827±0.0036	0.0326±0.0032	0.8367±0.0013	0.8361±0.0019	0.8361±0.0019	0.0470±0.0039
ALBERT w/ BNN	0.9265±0.0008	0.8862±0.0015	0.8862±0.0015	0.0411±0.0007	0.8339±0.0001	0.8338±0.0001	0.8338±0.0001	0.0527±0.0001
ALBERT w/ `CUE` (Ours)	0.9269±0.0020	0.8897±0.0044	0.8897±0.0044	0.0282±0.0018	0.8331±0.0003	0.8329±0.0003	0.8329±0.0003	**0.0338**±0.0007
DistilBERT (66M)	0.9287±0.0031	0.8886±0.0062	0.0441±0.0044	0.0393±0.0030	0.8067±0.0014	0.8059±0.0012	0.3737±0.0058	0.0376±0.0035
DistilBERT Label Smoothing	0.9264±0.0031	0.8841±0.0056	0.0716±0.0051	0.0353±0.0020	0.8049±0.0012	0.8040±0.0011	0.3994±0.0070	0.0319±0.0044
DistilBERT MC Dropout	0.9298±0.0018	0.8886±0.0025	0.0432±0.0051	0.0388±0.0020	0.8066±0.0020	0.8058±0.0019	0.3734±0.0121	0.0383±0.0066
DistilBERT w/ BNN	0.9315±0.0008	0.8931±0.0009	0.0413±0.0001	0.0406±0.0000	0.8059±0.0002	0.8052±0.0002	0.3673±0.0000	0.0424±0.0002
DistilBERT w/ `CUE` (Ours)	0.9295±0.0010	0.8911±0.0011	0.0900±0.0002	**0.0265**±0.0005	0.8058±0.0003	0.8051±0.0003	0.4600±0.0022	**0.0229**±0.0005
BERT (110M)	0.9296±0.0030	0.8871±0.0057	0.0523±0.0011	0.0335±0.0020	0.8286±0.0029	0.8281±0.0027	0.3361±0.0071	0.0321±0.0033
BERT Label Smoothing	0.9295±0.0042	0.8862±0.0074	0.0759±0.0024	**0.0289**±0.0061	0.8265±0.0016	0.8261±0.0017	0.3513±0.0116	0.0317±0.0043
BERT MC Dropout	0.9285±0.0027	0.8872±0.0048	0.0527±0.0026	0.0325±0.0037	0.8287±0.0030	0.8281±0.0028	0.3363±0.0071	0.0315±0.0032
BERT w/ BNN	0.9274±0.0008	0.8853±0.0011	0.0497±0.0001	0.0402±0.0006	0.3469±0.0194	0.1862±0.0179	0.9245±0.0835	0.1456±0.1111
BERT w/ `CUE` (Ours)	0.9259±0.0009	0.8850±0.0015	0.1031±0.0082	**0.0289**±0.0043	0.8283±0.0005	0.8277±0.0005	0.3665±0.0030	**0.0262**±0.0021
RoBERTa (125M)	0.9279±0.0033	0.8821±0.0062	0.0448±0.0043	0.0384±0.0050	0.8569±0.0043	0.8563±0.0044	0.2628±0.0127	0.0368±0.0074
RoBERTa Label Smoothing	0.9301±0.0024	0.8896±0.0051	0.0675±0.0026	0.0341±0.0049	0.8551±0.0022	0.8546±0.0020	0.3029±0.0082	**0.0255**±0.0033
RoBERTa MC Dropout	0.9305±0.0034	0.8919±0.0059	0.0514±0.0068	0.0315±0.0046	0.8586±0.0062	0.8581±0.0061	0.2516±0.0309	0.0403±0.0112
RoBERTa w/ BNN	0.9290±0.0006	0.8923±0.0005	0.0478±0.0002	0.0412±0.0007	0.8528±0.0001	0.8527±0.0001	0.2620±0.0000	0.0416±0.0002
RoBERTa w/ `CUE` (Ours)	0.9286±0.0005	0.8927±0.0008	0.0944±0.0004	**0.0313**±0.0033	0.8526±0.0003	0.8526±0.0003	0.3179±0.0003	0.0262±0.0004

Table 1: Results for `CUE` compare against baseline methods on four language models trained on four datasets. Values shown in parentheses indicate the model size. The highest entropy are underlined and the lowest ECE values are in bold.

where $\boldsymbol{e}_{ij}$ denotes the representation of the j-th input token. We assume that the adjacent layers in the transformer share similar representations. Then, $\boldsymbol{e}_i$ at layer L is:

$$\begin{aligned}\boldsymbol{e}_i^L &\propto \Sigma_{j=1}^{n} \exp(\langle \Sigma_{d=1}^{\dim} q_\phi(\boldsymbol{z}_{id}|\boldsymbol{e}_i) \cdot \boldsymbol{r}_{z_{id}}, \boldsymbol{e}_{ij} \rangle) \cdot \boldsymbol{e}_{ij} \\ &\propto \Sigma_{j=1}^{n} (\Sigma_{d=1}^{\dim} q_\phi(\boldsymbol{z}_{id}|\boldsymbol{e}_i) \cdot \langle \boldsymbol{r}_{z_{id}}, \boldsymbol{e}_{ij} \rangle) \cdot \boldsymbol{e}_{ij}\end{aligned} \tag{13}$$

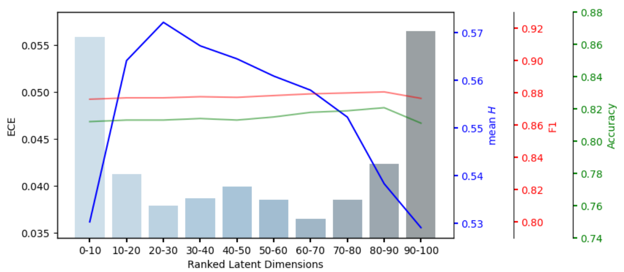
Therefore, the influence on prediction uncertainty changes $\Delta\mathcal{H}$ of the j-th token is decided by the generative probability of the encoder and the inner product $\langle \boldsymbol{r}_{z_{id}}, \boldsymbol{e}_{ij} \rangle$. However, seeking the optimal $\boldsymbol{z}_{id}$, by minimizing the reconstruction loss Eq. (5), is a typical Knapsack problem, which is an NP-complete problem. Hence, intuitively, we use greedy search to find a locally optimal solution by identifying the most influential latent dimensions of $\boldsymbol{z}_i$ first and then estimating the influential score for each token.

5 EXPERIMENTS

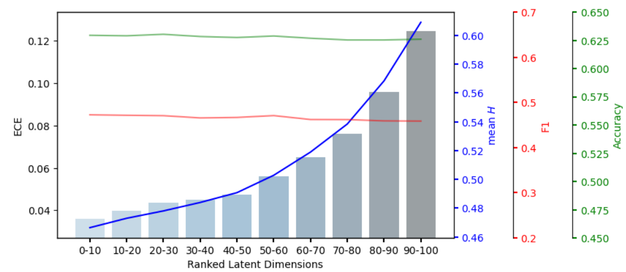
We first present the experimental setup followed by evaluation results.

Datasets We evaluate our proposed framework on four datasets for *linguistic acceptability classification*, *natural language inference*, and *emotion classification*.

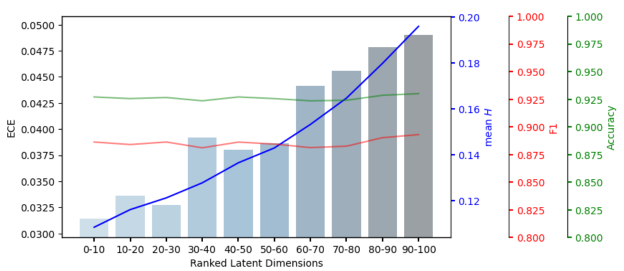
Baselines We compare our method with three baselines: Label Smoothing [Gupta et al., 2021], MC Dropout [Gal and Ghahramani, 2016] and Bayesian Neural Network (BNN). Label Smoothing and MC Dropout are implemented in PLMs and directly fine-tuned on the target datasets. The BNN works as a plug-in component, same as `CUE`, for which the base PLM encoding and the classification layer are firstly fine-tuned and then parameters are frozen for the plug-in



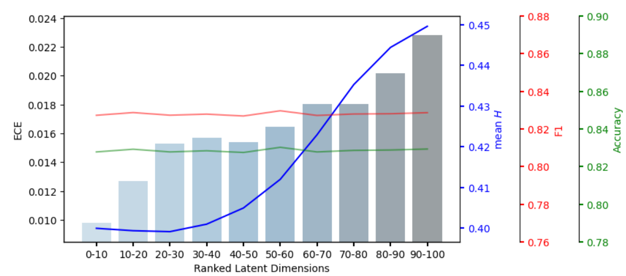
(a) BERT `CUE` on CoLA.



(b) BERT `CUE` on GoEmotions.



(c) BERT `CUE` on Emotion.



(d) BERT `CUE` on MultiNLI.

Figure 3: Evaluation results by removing latent dimensions. The x-axis represents the index of **removed** dimensions ranked by their relevance to $\Delta \boldsymbol{e}_i$, smaller index number indicates higher relevance. Histograms show the ECE scores after removing the corresponding latent dimensions. The blue curve shows the predictive entropy. The green and red curves show classification accuracy and F1, respectively.

layer training.

Evaluation Metrics Accuracy (Acc), macro-averaged F1 (F1), average entropy ($\mathcal{H}$), and Expected Calibration Error (ECE) are used as metrics for classification performance, uncertainty and model calibration measurement.

More details on dataset statistics, baseline setup, evaluation metrics and hyperparameter settings are in Supplementary Material Section 3.

5.1 OVERALL COMPARISON

Table 1 presents the performance of methods with four state-of-the-art PLMs, namely, BERT [Devlin et al., 2019], ALBERT [Lan et al., 2020], DistilBERT [Sanh et al., 2019] and RoBERTa [Liu et al., 2020b], as backbones. Our framework with a plug-in `CUE` module obtains the lowest ECE scores and highest average predictive entropy on all tasks and with different base model choices while maintaining a comparable level of Acc/F1 scores as the original model. Although the BNN model achieves the highest entropy with BERT on MultiNLI and CoLA dataset, we can observe a significant drop in its Acc/F1 scores. This indicates the BNN encoder hardly generates reliable perturbations that maintain predicted labels unchanged. Interestingly, while the classification performance of all compared models shows slight degradation with the injection of uncertainty into the PLMs, our framework achieves steady accuracy gains on the CoLA dataset.

5.2 EFFECTIVENESS OF THE UNCERTAINTY FEATURE IDENTIFICATION ALGORITHM

Results with Latent Dimension Removal As presented in §4.2, we can use the `CUE`'s reconstruction difference $\Delta \boldsymbol{e}_i$, to disentangle the most influential latent dimensions $\boldsymbol{z}_{id}$s which cause predictive uncertainty. Since each latent dimension is associated with an influential score, we can sort the latent dimensions accordingly. We speculate that by removing latent dimensions with higher influence scores, we should be able to observe a reduction in predictive uncertainty. As shown in Figure 3, we visualise the evaluation results by removing latent dimensions from $\boldsymbol{z}_i$ according to their relevance to $\Delta \boldsymbol{e}_i$ (the rank is shown on the x-axis) on BERT models. In our experiments, the latent vector $\boldsymbol{z}_i$ has 100 dimensions, we thus sort them into 10 bins in descending order based on their influential scores. In practice, the latent dimension removal is achieved by assigning 0 as the value of the dimension on $\boldsymbol{z}_i$ to create a modified latent variable $\boldsymbol{z}'_i$, new prediction is made with $\hat{y}'_i = g_{\boldsymbol{\eta}}(p_{\boldsymbol{\theta}}(\boldsymbol{z}'_i))$.

We can observe a remarkable increasing trend of ECE (the histograms) and average entropy (the blue curve) when removing the most influential latent dimensions of $\boldsymbol{z}_i$ on GoE-

Examples	Predicted	True
GoEmotions		
Despite having lived here for 10 years, I've never been to portillos, and given this, it's *somewhat unlikely* I start going now...	Disapproval 0.42→ 0.35	Neutral
Somehow I got banned for replying to a troll. The mods over there have *itchy trigger fingers*.	Disappointment 0.28 → 0.19	Disapproval
Boundaries. Have some boundaries. *Say no*. *Don't go*. This is frustrating to read, honestly. Don't do things that you hate doing.	Disgust 0.34 → 0.25	Fear
Emotion		
I were to *go overseas or cross the border* then I become a foreigner and will feel that way but never in my *beloved land*.	Joy 0.51 → 0.43	Love
I started feeling a little stressed about leaving on time and making sure we *got the getting ready pictures* I wanted but everything seemed to work out perfectly.	Sadness 0.59 → 0.40	Anger
I *wont* lie this week has been *abit of a difficult* week for me ive been feeling very stressed and anxious this week plus i think im coming down with the flu but it *has definately* helped me to appreciate the little things.	Sadness 0.64 → 0.44	Anger
MultiNLI		
P: There are no shares of a stock that might someday come back, just piles of options *as worthless as* those shares of cook's american business alliance. **H:** Cook's american business alliance caused shares of stock to come back.	Neutral 0.43 → 0.40	Contradiction
P: Until all *members* of our *society* are afforded that access, this promise of our government will continue to be *unfulfilled*. **H:** the government is flawed and *unfulfilled*.	Entailment 0.48 → 0.43	Neutral
P: Here you'll find the finest leather goods and of - the - moment fashions from all the predictable high - *priests* (valentino, armani, versace, gucci, missoni, etc.). A number of *classic* men's clothing meccas such as cucci (with a c), brioni, and battistoni are still *going strong*. **H:** You will find only the highest quality goods, be they *high - fashion icons* or top - *notch designer clothing* here.	Contradiction 0.35 → 0.33	Entailment

Table 2: Visualisation of token-level uncertainty interpretation. The '**Predicted**' column denotes the incorrect predictions made by the original fine-tuned model. Values below each predicted label denotes the predictive probability change after applying our framework. The '**True**' column denotes the gold-standard class labels. Italic text highlighted in blue are word tokens identified by UFI Algorithm that cause predictive uncertainty.

motions, Emotion and MultiNLI datasets. This indicates the top-ranked dimensions (smaller index number) contribute more to increase the predictive uncertainty and reducing the overconfidence prediction, while lower-ranked dimensions have less effect. Therefore, we can select the appropriate threshold for each dataset during token-level uncertainty identification by observing the trend of ECE and entropy. We also notice that across all datasets, removing any latent dimensions does not affect much the classification accuracy (the green curve) and macro-F1 (the red curve). However, we did not observe a similar trend of ECE and entropy on the CoLA dataset. We suspect this is due to a relatively simple setup in CoLA as it is only a binary classification task. For datasets with more classes, such as GoEmotion with 27 classes, the trend of ECE with latent dimension removal becomes more obvious. We also performed the same analysis and observed similar phenomena on other PLMs, DistilBERT, ALBERT and RoBERTa, in Supplementary Material Section 4.1.

Case Study of Token-Level Uncertainty Identification In this subsection, we demonstrate the effectiveness of our uncertainty identification algorithm by visualising the tokens that our framework finds contributing to predictive uncertainty. We present several examples in which our framework reduces overconfident predictions in Table 2. We only show the results with BERT as the base model due to page limits. Tokens coloured in blue are the influential tokens[4] identified by the UFI Algorithm.

For emotion classification, we found classifiers tend to be confused by idioms or phrases carrying emotions different from the true emotion labels. For example, the second sentence in GoEmotion contains a metaphorical phrase, '*itchy trigger finger*', making it a tricky case for emotion classification. We conducted additional experiments in which we substituted the phrase "itchy trigger finger" with either the `[MASK]` token or commonly used words to express the same meaning. In both cases, the model uncertainty is reduced by replacing the original phrase with the mask tokens leading to label switching. Replacing the identified phrase with more commonly-used words increases the predictive probability and leads to a more confident prediction. These results verify the validity of our approach for identifying words/phrases causing predictive uncertainties. The first and last sentences in GoEmotion and also the last sentence in Emotion contain phrases which are somewhat more closely related to the incorrectly predicted labels than the true labels, confusing

[4]For words split into subword tokens, we take the average importance score of the constituent subword tokens.

the classifier to generate wrong predictions. Tokenisation may also cause a problem. For example, the word '*beloved*' in the first sentence in Emotion is split into three parts after tokenisation, making it difficult for the classifier to recognise the 'Love' emotion. For the natural language inference task, we found classifiers tend to make overconfident predictions when the same words are found in both premise and hypothesis. For examples, the second instance in MultiNLI has the word '*unfulfilled*' in both its premise and hypothesis. This leads to the wrong prediction of 'Entailment'. In the last instance, the classifier misunderstood that the '*classic men's clothing*' contradicts with '*high - fashion icons*' and thus failed to recognise the 'Entailment' relation. Nevertheless, in all these cases, our proposed framework managed to increase the predictive entropy by reducing the confidence of predictions, alleviating the overconfidence problem.

We provide further experimental results and the ablation study, including stability of various additional training loss terms and latent space orthogonality in Supplementary Material Section 4.2.

6 CONCLUSION

In this paper, we have proposed a new framework `CUE` for uncertainty interpretation of PLM classifiers. By comparing our method with previous solutions, we show that `CUE` can achieve lower expected calibration errors across four datasets. In some cases, it can also mitigate the confidence of previously wrong predictions. Further experiments and case studies demonstrate `CUE` is effective in identifying tokens/latent dimensions that could potentially cause predictive uncertainty. Our work sheds light on a new direction of uncertainty interpretation for PLMs in various NLP tasks.

Acknowledgements

This work was supported in part by the UK Engineering and Physical Sciences Research Council (grant no. EP/T017112/2, EP/V048597/1, EP/X019063/1). YH is supported by a Turing AI Fellowship funded by the UK Research and Innovation (grant no. EP/V020579/2). The authors would like to thank Yuxiang Zhou, Hanqi Yan and Xingwei Tan for their invaluable feedback on this paper.

References

C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural network. In *Proceedings of the 32nd ICML*, pages 1613–1622. PMLR, 2015.

G. Brunner, Y. Liu, D. Pascual, O. Richter, M. Ciaramita, and R. Wattenhofer. On identifiability in transformers. In *ICLR*, 2020.

D. Card, C. Tan, and N. A. Smith. Neural models for documents with metadata. In *Proceedings of the 56th Annual Meeting of the ACL*, pages 2031–2040, July 2018. doi: 10.18653/v1/P18-1189.

M. R. Cervera, R. Dätwyler, F. D'Angelo, H. Keurti, B. F. Grewe, and C. Henning. Uncertainty estimation under model misspecification in neural network regression. *CoRR*, abs/2111.11763, 2021.

K. Clark, U. Khandelwal, O. Levy, and C. D. Manning. What does BERT look at? An Analysis of BERT's Attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Aug. 2019. doi: 10.18653/v1/W19-4828.

S. Desai and G. Durrett. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on EMNLP*, pages 295–302, Nov. 2020. doi: 10.18653/v1/2020.emnlp-main.21.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the NACCL*, pages 4171–4186, June 2019. doi: 10.18653/v1/N19-1423.

Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd ICML*, pages 1050–1059. PMLR, 2016.

C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th ICML*, pages 1321–1330. PMLR, 2017.

A. Gupta, G. Kvernadze, and V. Srikumar. Bert & family eat word salad: Experiments with text understanding. *AAAI*, 35(14):12946–12954, May 2021. doi: 10.1609/aaai.v35i14.17531.

J. He, X. Zhang, S. Lei, Z. Chen, F. Chen, A. Alhamadani, B. Xiao, and C. Lu. Towards more accurate uncertainty estimation in text classification. In *Proceedings of the 2020 Conference on EMNLP*, pages 8362–8372, Nov. 2020. doi: 10.18653/v1/2020.emnlp-main.671.

J. Heiss, J. Weissteiner, H. Wutte, S. Seuken, and J. Teichmann. Nomu: Neural optimization-based model uncertainty, 2023.

S. Kim, J. Yi, E. Kim, and S. Yoon. Interpretation of NLP models through input marginalization. In *Proceedings of the 2020 Conference on EMNLP*, pages 3154–3167, Nov. 2020. doi: 10.18653/v1/2020.emnlp-main.255.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. In Y. Bengio and Y. LeCun, editors, *2nd ICLR, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, volume 30. Curran Associates, Inc., 2017.

Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. Albert: A lite bert for self-supervised learning of language representations. In *ICLR*, 2020.

J. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax Weiss, and B. Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. In *NeurIPS*, volume 33, pages 7498–7512, 2020a.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach, 2020b.

A. Malinin and M. Gales. Uncertainty estimation in autoregressive structured prediction. In *ICLR*, 2021.

A. Malinin and M. J. F. Gales. Predictive uncertainty estimation via prior networks. In *NeurIPS*, 2018.

D. Mareček and R. Rosa. From balustrades to pierre vinken: Looking for syntax in transformer self-attentions. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 263–275, Aug. 2019. doi: 10.18653/v1/W19-4827.

X. Qiu, E. Meyerson, and R. Miikkulainen. Quantifying point-prediction uncertainty in neural networks via residual estimation with an i/o kernel. In *ICLR*, 2020.

V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019.

A. Shelmanov, E. Tsymbalov, D. Puzyrev, K. Fedyanin, A. Panchenko, and M. Panov. How certain is your Transformer? In *Proceedings of the 16th Conference of the EACL*, pages 1833–1840, Apr. 2021. doi: 10.18653/v1/2021.eacl-main.157.

Y. Shen, W. Chen, and H. Jin. Modeling token-level uncertainty to learn unknown concepts in SLU via calibrated dirichlet prior RNN. *CoRR*, abs/2010.08101, 2020.

T. J. Sullivan. *Introduction to uncertainty quantification*, volume 63. Springer, 2015.

K. Sun and A. Marasović. Effective attention sheds light on interpretability. In *Findings of the ACL-IJCNLP 2021*, pages 4126–4135, Aug. 2021. doi: 10.18653/v1/2021.findings-acl.361.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, volume 30. Curran Associates, Inc., 2017.

A. Vazhentsev, G. Kuzmin, A. Shelmanov, A. Tsvigun, E. Tsymbalov, K. Fedyanin, M. Panov, A. Panchenko, G. Gusev, M. Burtsev, M. Avetisian, and L. Zhukov. Uncertainty estimation of transformer predictions for misclassification detection. In *Proceedings of the 60th Annual Meeting of the ACL*, pages 8237–8252, May 2022.

I. Vulić, E. M. Ponti, R. Litschko, G. Glavaš, and A. Korhonen. Probing pretrained language models for lexical semantics. In *Proceedings of the 2020 Conference on EMNLP*, pages 7222–7240, Nov. 2020. doi: 10.18653/v1/2020.emnlp-main.586.

Y. Xiao and W. Y. Wang. Quantifying uncertainties in natural language processing tasks. In *Proceedings of the 33rd AAAI*. AAAI Press, 2019. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.33017322.

L. Yang, J. Li, P. Cunningham, Y. Zhang, B. Smyth, and R. Dong. Exploring the efficacy of automatically generated counterfactuals for sentiment analysis. In *ACL 2021*, pages 306–316, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.26.

Y. Zhou and V. Srikumar. DirectProbe: Studying representations without classifiers. In *Proceedings of the 2021 Conference of NAACL*, pages 5070–5083, June 2021. doi: 10.18653/v1/2021.naacl-main.401.