Uncertainty Estimation of Transformer Predictions for Misclassification Detection

Artem Vazhentsev[1,2] ♢, Gleb Kuzmin[1,6] ♢, Artem Shelmanov[1,7] ♢, Akim Tsvigun[1,4], Evgenii Tsymbalov[2], Kirill Fedyanin[2], Maxim Panov[2], Alexander Panchenko[2], Gleb Gusev[1,3,5], Mikhail Burtsev[1,3], Manvel Avetisian[1,5], and Leonid Zhukov[1,4]

[1]AIRI, [2]Skoltech, [3]MIPT, [4]HSE, [5]Sber AI Lab, [6]FRC CSC RAS,
[7]ISP RAS Research Center for Trusted Artificial Intelligence

{vazhentsev, kuzmin, shelmanov, tsvigun, gusev, burtsev, manvel, zhukov}@airi.net
{evgenii.tsymbalov, m.panov, k.fedyanin, a.panchenko}@skoltech.ru

Abstract

Uncertainty estimation (UE) of model predictions is a crucial step for a variety of tasks such as active learning, misclassification detection, adversarial attack detection, out-of-distribution detection, etc. Most of the works on modeling the uncertainty of deep neural networks evaluate these methods on image classification tasks. Little attention has been paid to UE in natural language processing. To fill this gap, we perform a vast empirical investigation of state-of-the-art UE methods for Transformer models on misclassification detection in named entity recognition and text classification tasks and propose two computationally efficient modifications, one of which approaches or even outperforms computationally intensive methods[1].

1 Introduction

Machine learning methods are naturally prone to errors as they typically have to deal with ambiguous and incomplete data during both training and inference. Unreliable predictions hinder the application of these methods in domains, where the price of mistakes is very high, such as clinical medicine. Even in more error-tolerant domains and tasks, such as intent recognition in general-purpose chatbots, one would like to achieve a better trade-off between expressiveness of a model and its computational performance during inference.

Since mistakes are inevitable, it is crucial to understand whether model predictions can be trusted or not and abstain from unreliable decisions. Uncertainty estimation (UE) of model predictions aims to solve this task. Ideally, uncertain instances should correspond to erroneous objects and help in *misclassification detection*. Besides misclassification detection, UE is a crucial component for active learning (Settles, 2009), adversarial attack detection (Lee et al., 2018), detection of out-of-distribution (OOD) instances (Van Amersfoort et al., 2020), etc.

Some classical machine learning models, e.g. Gaussian processes (Rasmussen, 2003), have built-in UE capabilities. Modern deep neural networks (DNNs) usually take advantage of a softmax layer, which output can be considered as a prediction probability and be used for UE. However, the softmax probabilities are usually unreliable and produce overconfident predictions (Guo et al., 2017). Some previously proposed techniques such as deep ensemble (Lakshminarayanan et al., 2017) are known for producing good UE scores but require a large additional memory footprint for storing several versions of weights and multiply an amount of computation for conducting several forward passes. Reliable UE of DNN predictions that does not introduce high computational overhead is an open research question (Van Amersfoort et al., 2020).

In this work, we investigate methods for UE of DNNs based on the Transformer architecture (Vaswani et al., 2017) in misclassification detection. We consider two of the most common NLP tasks: text classification and named entity recognition (NER). The latter has been overlooked in the literature on UE. To our knowledge, this work is the first to consider UE for NER.

We propose two novel computationally cheap methods for UE of Transformer predictions. The first method is the modification of the Monte Carlo dropout with determinantal point process sampling of dropout masks (Shelmanov et al., 2021). We introduce an additional step for making masks more diverse, which helps to

[1]The code for experiments is available online at https://github.com/AIRI-Institute/uncertainty_transformers

♢ Equal contribution, corresponding authors

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 8237 - 8252

achieve substantial improvements and approach the performance of computationally-intensive methods on NER. The second method leverages Mahalanobis distance (Lee et al., 2018) but also adds a spectral normalization of the weight matrix in the classification layer (Liu et al., 2020). This method achieves the best results on most of the datasets and even outperforms computationally-intensive methods. We also investigate recently proposed regularization techniques in combination with other UE methods. The contributions of this paper are the following:

- We propose two novel computationally cheap modifications of UE methods for Transformer models. The method based on Mahalanobis distance with spectral normalization approaches or even outperforms strong computationally intensive counterparts.
- This work is the first to investigate UE methods on the NER task.
- We conduct an extensive empirical evaluation, in which we investigate recently proposed regularization techniques in combination with other UE methods.

2 Related Work

It is well known that reliable uncertainty scores can be obtained simply by constructing an ensemble of decorrelated neural networks (*deep ensemble*) (Lakshminarayanan et al., 2017). However, such a straightforward approach is coupled with substantial computational and memory overhead during training an ensemble, performing inference of all its components, and storing multiple versions of weights. This overhead is a serious obstacle to deploying ensemble-based uncertainty estimation methods in practice.

Uncertainty estimation is a built-in capability of Bayesian neural networks (Blundell et al., 2015). However, such models have similar issues as ensembles and also require special training procedures. Recently, it was shown by Gal and Ghahramani (2016) that dropout, a well-known regularization technique, is formally equivalent to approximate variational inference in a deep Gaussian process if it is activated during prediction. This method, known as Monte Carlo (MC) dropout, uses the approximating variational distribution with Bernoulli variables related to network units. MC dropout does not impose any overhead during training, introduces no additional parameters, and thus does not require any additional memory. The main disadvantage of this method is that it usually requires many forward-pass samplings for approximating predictive posterior, which makes it also computationally expensive.

Recently, many works have investigated the approximate Bayesian inference for neural networks using deterministic approaches: Lee et al. (2018); Liu et al. (2020); Van Amersfoort et al. (2020); Mukhoti et al. (2021); Shen et al. (2021), etc. These methods do not introduce notable overhead for inference, storing weights, and usually require compatible training time. However, most of the research in this area is accomplished for computer vision tasks.

For text classification, a series of works investigates UE methods for the OOD detection task (Liu et al., 2020; Podolskiy et al., 2021; Zeng et al., 2021; Hu and Khan, 2021). In this work, we focus on a more challenging task – misclassification detection. While OOD detection requires to model only the epistemic uncertainty inherent to the model and caused by a lack of training data, misclassification detection also requires to model aleatoric uncertainty caused by noise and ambiguity in data (Mukhoti et al., 2021). We consider recently proposed methods in this area that are evaluated in text processing.

Three recent works propose techniques for misclassification detection based on an additive regularization of a training loss function. Zhang et al. (2019) suggest adding a penalty that reduces the Euclidean distance between training instances of the same class and increases the distance between instances of different classes. He et al. (2020) suggest using two components in the loss function that reduce the difference between outputs from two versions of a model initialized with different weights. They also use mix-up (Thulasidasan et al., 2019) to generate additional training instance representations that help to capture aleatoric uncertainty, self-ensembling, MC dropout, and a distinctiveness score to measure the epistemic uncertainty. Xin et al. (2021) introduce a regularizer that penalizes overconfident instances with high loss. In another recent work, Shelmanov et al. (2021) propose to combine MC dropout with a Determinantal Point Process (DPP) to improve the diversity of predictions by considering the correlations between neurons and sampling the

diverse neurons for activation in a dropout layer.

In this work, we conduct a systematic empirical investigation of UE methods on NLP tasks. We evaluate combinations of methods that have not been tested before and propose two modifications, one of which achieves the best results among computationally cheap methods. The previous work focuses on text classification tasks, while this work is the first to investigate UE also for NER.

3 Background and Methods

In this section, we describe the baselines and propose novel uncertainty estimation techniques.

3.1 Softmax Response

Softmax Response (SR) (Geifman and El-Yaniv, 2017) is a trivial baseline for UE that uses the probabilities generated via the output softmax layer of the neural network. SR is based on the maximum probability $p(y|x)$ over classes $y = c \in C$. The smaller this probability is, the more uncertain model is:

$$u_{\text{SR}}(x) = 1 - \max_{c \in C} p(y = c|x). \quad (1)$$

3.2 Monte Carlo Dropout

Standard Monte Carlo Dropout (MC Dropout) Consider we have conducted T stochastic forward passes with activated dropout. In this work, we use the following ways to quantify uncertainty with methods based on MC dropout:

- Sampled maximum probability (SMP) is:

$$u_{\text{SMP}} = 1 - \max_{c \in C} \frac{1}{T} \sum_{t=1}^{T} p_t^c, \quad (2)$$

  where p_t^c is the probability of the class c for the t-th stochastic forward pass.

- Probability variance (PV; Gal et al. (2017); Smith and Gal (2018)) is:

$$u_{\text{PV}} = \frac{1}{C} \sum_{c=1}^{C} \left(\frac{1}{T} \sum_{t=1}^{T} (p_t^c - \overline{p^c})^2 \right), \quad (3)$$

  where $\overline{p^c} = \frac{1}{T} \sum_t p_t^c$ is the probability for a class c averaged across T stochastic forward passes.

- Bayesian active learning by disagreement (BALD; Houlsby et al. (2011)) is:

$$u_{\text{BALD}} = -\sum_{c=1}^{C} \overline{p^c} \log \overline{p^c} + \frac{1}{T} \sum_{c,t} p_t^c \log p_t^c. \quad (4)$$

The two former techniques are specifically designed for estimation of the epistemic (model) uncertainty arising from the lack of knowledge and ignore the aleatoric uncertainty related to ambiguity and noise in the data, while the latter method can be seen as a measure of total uncertainty (Malinin and Gales, 2018).

Transformers contain multiple dropout layers (after the embedding layer, in each attention head, and before the last classification layer). It is shown in previous work that the standard MC dropout outperforms the baseline SR only when all dropout layers are activated in a model (Shelmanov et al., 2021). Therefore, we follow this setting for experiments in this work. We note that due to activating all dropout layers, multiple stochastic predictions are required for the whole network, which introduces a large computational overhead.

Similar UE scores are used in deep ensemble (Lakshminarayanan et al., 2017), where instead of multiple stochastic predictions we train and infer several model versions with different sets of weights.

Diverse Determinantal Point Process Monte Carlo Dropout (DDPP MC dropout) (Ours) Determinantal point processes (DPPs; Kulesza and Taskar (2012)) are used for sampling a subset of diverse objects from a given set. Recently, Shelmanov et al. (2021) have combined the MC dropout with a determinantal point process (DPP) for sampling neurons in a dropout layer and demonstrated that using stochasticity in the last dropout layer (in a classification head of Transformer) only is enough to improve upon SR in misclassification detection. This method is less computationally expensive than the standard MC dropout since it requires multiple stochastic predictions only for the top classification layer of the network with a small number of parameters, while all other layers are inferred only once.

Consider the similarity matrix C_h between neurons of the h-th hidden layer (in particular, we use a correlation matrix between output values of neurons on the training set). Then one can construct the DPP-based dropout masks M_h^{DPP} using C_h as a likelihood kernel for the DPP: $M_h^{DPP} \sim DPP(C_h)$. That gives the following probability to select a set S of activations on the layer h:

$$P\left[M_h^{DPP} = S\right] = \frac{\det(C_h^S)}{\det(C_h + I)}, \quad (5)$$

where C_h^S is the square submatrix of C_h obtained by keeping only rows and columns indexed by the sample S.

In this work, we improve this method by increasing the diversity of the sampled DPP masks. After multiple dropout masks are pre-generated via DPP in the inference step as in the original DPP MC dropout, we make an additional step, in which we select a diverse set of masks from this pre-generated pool using one of two strategies:

- **DDPP (+DPP)**: We sample a set of "diverse" masks that activate different sets of neurons. For this purpose, we apply DPP sampling again to the pool of pre-generated masks. As a similarity kernel in this step, we use an RBF-similarity matrix of mask vectors.
- **DDPP (+OOD)**: We sample a set of masks that generate diverse predictions. For this purpose, we select the masks that yield the highest PV scores on the given OOD dataset.

After a new set of T masks is selected, we use them as in the standard MC dropout to obtain stochastic predictions. Increasing the diversity of masks in the proposed modification is motivated by the finding of Jain et al. (2020) that improving the diversity of elements in an ensemble leads to better uncertainty estimates.

We note that in masks generated with DPP, usually, less than 50% of neurons are activated, which makes predictions poorly calibrated. To mitigate this problem, for each constructed mask, we perform a temperature-scaling calibration (Guo et al., 2017) using a held-out dataset.

3.3 Deterministic Uncertainty Estimation

Spectral-normalized Neural Gaussian Process (SNGP) Liu et al. (2020) suggest replacing the typical dense output layer of a network with a layer that implements a Gaussian process with an RBF kernel, whose posterior variance at a given instance is characterized by its L_2 distance from the training data in the hidden vector space constructed by underlying layers of a network. The authors propose an approximation based on random Fourier feature expansion, which enables end-to-end training and makes the inference feasible.

However, this method requires hidden representations to be distance-preserving in order to make it work. While the distance between instances in the hidden space does not always have a meaningful correspondence to the distance in the input space, authors prove that to keep hidden representations distance-preserving, the transformation should satisfy the bi-Lipschitz condition. For ResNets (He et al., 2016), this requirement is satisfied if weight matrices for the nonlinear residual blocks have a spectral norm (i.e., the largest singular value) bounded from above by a constant. Therefore, to enforce the aforementioned Lipschitz constraint, they apply a spectral normalization (SN) on weight matrices. For Transformers, they normalize the matrix of the penultimate classification layer only.

Mahalanobis Distance (MD) Mahalanobis distance is a generalisation of the Euclidean distance, which takes into account the spreading of instances in the training set along various directions in a feature space. Lee et al. (2018) suggest estimating uncertainty by measuring the Mahalanobis distance between a test instance and the closest class-conditional Gaussian distribution:

$$u_{\text{MD}} = \min_{c \in C} (h_i - \mu_c)^T \Sigma^{-1} (h_i - \mu_c), \quad (6)$$

where h_i is a hidden representation of a i-th instance, μ_c is a centroid of a class c, and Σ is a covariance matrix for hidden representations of training instances.

Recently, the Mahalanobis distance has been adopted for out-of-distribution detection with Transformer networks by Podolskiy et al. (2021).

Mahalanobis Distance with Spectral-normalized Network (MD SN) (Ours) Since the UE method based on the Mahalanobis distance utilizes the idea of a proximity of a tested instance hidden representation to the training distribution, we expect this method to benefit from distance-preserving representations. Therefore, we propose the modification of the method of Lee et al. (2018) and Podolskiy et al. (2021) that enforces the bi-Lipschitz constraints on transformation implemented by the network. We perform spectral normalization of the weight matrix of the linear layer in the classification head of Transformer as it is suggested in SNGP (Liu et al., 2020). At each training step, a spectral norm ν is estimated using the power iteration method $\nu = \|W\|_2$, and a normalized weight matrix is obtained: $\bar{W} = \frac{W}{\nu}$. At the inference step, hidden representations are calculated using the normalized

matrix $\tilde{h}(x) = \tilde{W}x + b$ and are used for computing the Mahalanobis distance.

3.4 Training Loss Regularization

Additive regularization is another approach to improving UE of neural network predictions. Usually, the training loss combines the original task-specific loss L_{task} (e.g. cross-entropy) and a regularization component L_{reg} that facilitates producing better calibrated UEs:

$$L = L_{task} + \lambda L_{reg}, \tag{7}$$

where λ is a hyperparameter that controls the regularization strength.

The positive side of such techniques is that, besides SR, they can be used to improve other methods like MC dropout and deterministic methods. The drawback is that regularization affects the training procedure and can decrease the model quality.

Confident Error Regularizer (CER) Xin et al. (2021) propose a regularizer that adds a penalty for an instance with a bigger loss than other instances and, at the same time, bigger confidence:

$$L_{reg} = \sum_{i,j=1}^{k} \Delta_{i,j} \mathbb{1}[e_i > e_j], \tag{8}$$

$$\Delta_{i,j} = \max\{0, \max_c p_i^c - \max_c p_j^c\}^2, \tag{9}$$

where k is the number of instances in a batch and e_i is an error of the i-th instance: e_i is 1 if the prediction of the classifier matches the true label, and e_i is 0 otherwise. The authors evaluate this type of regularization only in conjunction with the SR baseline.

Metric Regularizer Zhang et al. (2019) propose a regularizer that aims to shorten the intra-class distance and enlarge the inter-class distance:

$$L_{reg} = \sum_{c=1}^{C} \Big\{ L_{intra}(c) + \varepsilon \sum_{k \neq c} L_{inter}(c,k) \Big\}, \tag{10}$$

$$L_{intra}(c) = \frac{2}{|S_c|^2 - |S_c|} \sum_{i,j \in S_c, i<j} D(h_i, h_j), \tag{11}$$

$$L_{inter}(c,k) = \frac{1}{|S_c| \cdot |S_k|} \sum_{i \in S_c, j \in S_k} [\gamma - D(h_i, h_j)]_+, \tag{12}$$

$$D(r_i, r_j) = \frac{1}{d} ||h_i - h_j||_2^2, \tag{13}$$

where h_i is a feature representation of an instance i from a penultimate layer of a model with a dimension d, S_c is the set of instances from class c, $|S_c|$ is the number of elements in S_c, ε and γ are positive hyperparameters, $[x]_+ = max(0, x)$.

4 Experimental Setup

In the experiments, we train a model on a given dataset and perform inference on a separate test set to compute both predictions and UE scores u. We are interested in how the scores correlate with the mistakes $\tilde{e}$ of the model on the test set. For text classification, mistakes are computed in the following way:

$$\tilde{e}_i = \begin{cases} 1, & y_i \neq \hat{y}_i, \\ 0, & y_i = \hat{y}_i, \end{cases} \tag{14}$$

where y_i is a true label, $\hat{y}_i$ is a predicted label.

For NER, we use two evaluation options: token-level and sequence-level. For the token-level evaluation, individual tokens are considered as separate instances as in the text classification. For the sequence-level evaluation, mistakes are computed in the following way:

$$\tilde{e}_i = \begin{cases} 1, & \exists j \in \{1, \ldots, n\}, \ y_{ij} \neq \hat{y}_{ij}, \\ 0, & \forall j \in \{1, \ldots, n\}, \ y_{ij} = \hat{y}_{ij}, \end{cases} \tag{15}$$

where n is a sequence length, y_{ij} is a true label, $\hat{y}_{ij}$ is a predicted label of a j-th token in a sequence. In the sequence-level evaluation, UE of a sequence is aggregated from UEs of tokens by taking maximum (for MD methods) or by summation (for others).

4.1 Metrics

El-Yaniv and Wiener (2010) suggest evaluating the quality of UE using the **area under the risk coverage curve (RCC-AUC)**. The risk coverage curve demonstrates the cumulative sum of loss due to misclassification (cumulative risk) depending on the uncertainty level used for rejection of predictions. The lower area under this curve indicates better quality of the UE method.

Xin et al. (2021) propose a **reversed pair proportion (RPP)** metric. They note that instances with higher confidence should have a lower loss l. RPP measures how far the uncertainty estimator $\tilde{u}$ is to ideal, given the labeled dataset of size n:

$$RPP = \frac{1}{n^2} \sum_{i,j=1}^{n} \mathbb{1}[\tilde{u}(x_i) > \tilde{u}(x_j), l_i < l_j]. \tag{16}$$

Method	Reg. Type	UE Score	MRPC		SST-2		CoLA		CoNLL-2003 (token level)		CoNLL-2003 (seq. level)	
			RCC-AUC ↓	RPP ↓	RCC-AUC ↓	RPP ↓	RCC-AUC ↓	RPP ↓	RCC-AUC ↓	RPP ↓	RCC-AUC ↓	RPP ↓
MC	-	PV	13.97±1.16	1.68±0.09	12.90±1.92	0.82±0.11	44.35±4.90	2.06±0.16	6.32±1.66	0.10±0.02	16.05±3.78	1.93±0.43
MC	-	BALD	14.21±1.04	1.69±0.09	12.98±1.87	0.82±0.10	45.06±4.90	2.08±0.17	6.44±1.86	0.10±0.02	16.28±4.00	1.96±0.45
MC	-	SMP	14.38±2.07	1.76±0.19	14.00±2.20	0.91±0.15	42.95±5.98	2.01±0.15	**6.04±1.03**	**0.09±0.02**	**15.79±3.34**	1.80±0.35
MC	CER	PV	**12.82±1.89**	**1.60±0.13**	12.18±1.20	0.80±0.10	46.84±9.19	2.11±0.23	6.92±1.22	0.10±0.02	17.05±3.14	1.91±0.36
MC	CER	BALD	12.89±1.89	**1.60±0.13**	12.39±1.23	0.81±0.09	47.34±8.30	2.14±0.24	7.16±1.15	0.11±0.02	17.25±3.05	1.93±0.35
MC	CER	SMP	12.91±2.15	1.67±0.15	12.22±1.31	0.82±0.09	46.10±11.07	2.05±0.22	6.69±1.38	0.10±0.02	16.81±1.61	1.81±0.14
MC	metric	PV	14.21±1.95	1.73±0.23	12.28±1.77	0.80±0.11	42.35±0.69	2.04±0.07	6.69±0.89	0.10±0.01	17.17±1.90	1.93±0.31
MC	metric	BALD	14.55±2.31	1.73±0.23	**12.08±1.79**	**0.79±0.10**	43.76±0.55	2.08±0.07	6.91±1.02	0.10±0.01	17.47±1.85	1.98±0.30
MC	metric	SMP	13.39±1.19	1.72±0.20	13.55±1.65	0.90±0.14	**40.88±1.25**	**2.01±0.09**	6.30±0.98	0.10±0.01	16.81±1.40	**1.80±0.23**
DDPP (+DPP) (ours)	-	PV	22.30±7.15	2.58±0.65	16.70±1.38	1.12±0.12	49.75±3.96	2.44±0.29	6.12±0.71	0.10±0.01	16.78±2.44	1.93±0.20
DDPP (+DPP) (ours)	-	BALD	23.08±7.00	2.63±0.63	16.08±2.37	1.05±0.18	49.59±5.40	2.48±0.31	6.39±0.64	0.10±0.01	21.53±4.77	2.63±0.45
DDPP (+DPP) (ours)	-	SMP	21.79±7.72	2.57±0.68	17.55±3.03	1.19±0.23	47.86±5.51	2.39±0.31	6.08±0.62	0.10±0.01	17.71±2.77	2.05±0.23
DDPP (+DPP) (ours)	CER	PV	15.12±2.27	2.03±0.24	13.56±1.37	0.91±0.14	54.51±8.80	2.58±0.22	6.98±0.98	0.11±0.02	19.44±1.15	2.13±0.17
DDPP (+DPP) (ours)	CER	BALD	15.94±3.77	2.07±0.36	14.87±2.22	0.96±0.13	55.11±7.42	2.61±0.31	7.90±1.95	0.12±0.01	26.20±6.41	3.11±0.56
DDPP (+DPP) (ours)	CER	SMP	14.75±1.43	2.02±0.16	14.47±1.63	0.99±0.11	54.01±9.79	2.55±0.18	6.91±1.13	0.11±0.02	20.66±1.53	2.31±0.08
DDPP (+DPP) (ours)	metric	PV	19.51±3.40	2.47±0.28	15.79±1.67	1.07±0.14	43.82±1.82	2.17±0.14	7.33±1.53	0.12±0.02	18.93±2.09	2.11±0.25
DDPP (+DPP) (ours)	metric	BALD	20.54±4.72	2.52±0.34	15.48±1.81	1.03±0.08	43.95±1.68	2.17±0.12	8.01±2.08	0.13±0.03	22.44±4.78	2.67±0.49
DDPP (+DPP) (ours)	metric	SMP	18.45±2.88	2.41±0.26	16.78±3.43	1.14±0.26	43.61±1.61	2.16±0.11	6.92±1.32	0.11±0.02	19.11±2.14	2.16±0.22
DDPP (+OOD) (ours)	-	PV	22.73±7.45	2.65±0.59	19.05±2.95	1.29±0.23	51.11±12.03	2.37±0.34	6.32±0.72	0.10±0.01	16.75±2.31	1.94±0.21
DDPP (+OOD) (ours)	-	BALD	23.85±8.39	2.69±0.58	18.27±3.05	1.22±0.23	52.59±12.08	2.42±0.34	6.59±0.69	0.11±0.01	20.56±3.09	2.50±0.26
DDPP (+OOD) (ours)	-	SMP	22.31±7.80	2.60±0.65	19.86±3.83	1.36±0.29	50.14±9.73	2.32±0.30	6.09±0.67	0.10±0.01	17.76±2.75	2.06±0.23
DDPP (+OOD) (ours)	CER	PV	14.83±1.42	2.05±0.17	14.98±1.36	1.01±0.09	59.14±11.27	2.56±0.24	7.08±1.37	0.11±0.02	19.66±1.25	2.17±0.15
DDPP (+OOD) (ours)	CER	BALD	15.03±1.85	2.08±0.24	14.37±2.22	0.96±0.14	57.48±9.37	2.54±0.26	7.41±1.29	0.12±0.02	25.30±3.36	3.00±0.24
DDPP (+OOD) (ours)	CER	SMP	14.34±1.15	1.99±0.16	15.88±1.96	1.08±0.13	59.32±11.86	2.53±0.20	6.88±1.24	0.11±0.02	21.06±1.96	2.35±0.14
DDPP (+OOD) (ours)	metric	PV	19.03±3.97	2.41±0.34	17.75±5.20	1.10±0.17	48.54±11.38	2.23±0.24	6.92±1.32	0.11±0.02	18.36±1.90	2.05±0.26
DDPP (+OOD) (ours)	metric	BALD	19.33±4.78	2.41±0.40	16.71±7.13	1.02±0.20	49.31±11.87	2.24±0.25	7.21±1.49	0.11±0.02	21.35±4.47	2.54±0.45
DDPP (+OOD) (ours)	metric	SMP	18.55±3.06	2.42±0.27	17.08±3.78	1.14±0.26	43.67±1.77	2.15±0.11	6.71±1.18	0.10±0.02	19.01±2.30	2.16±0.25
SR	CER	MP	14.62±1.62	2.02±0.19	14.56±2.14	1.00±0.14	56.97±9.69	2.53±0.15	6.84±1.41	0.11±0.02	21.31±1.63	2.49±0.25
SR	metric	MP	18.39±2.94	2.40±0.27	16.90±3.12	1.16±0.24	44.54±2.11	2.22±0.15	6.51±1.07	0.10±0.02	20.32±1.68	2.32±0.23
SR (baseline)	-	MP	22.32±8.08	2.58±0.65	17.93±3.84	1.22±0.28	49.48±3.71	2.35±0.25	6.08±0.62	0.10±0.01	18.81±3.35	2.21±0.29

Table 1: Results for methods based on MC dropout and regularization techniques (ELECTRA model). The best results are shown in bold, the best results for each method are underlined.

This metric has an upper bound of 1; for convenience, the reported values are multiplied by 100. Similar to Xin et al. (2021), for both metrics, l is an indicator loss function.

We conduct each experiment six times with different random seeds, obtaining the corresponding metric values, and report their mean and standard deviation.

We also present the results using the **accuracy rejection curve**. This curve is drawn by varying the rejection uncertainty level (horizontal axis) and presenting the corresponding accuracy obtained when all rejected instances are labeled with an oracle (vertical axis). This emulates the work of a human expert in conjunction with a machine learning system. The higher the curve, the smaller amount of labor is needed to achieve a certain level of performance and the better is the UE method. A similar evaluation approach in a table form is used in (Zhang et al., 2019). A similar curve but without oracle labeling is used in (Lakshminarayanan et al., 2017; Filos et al., 2019).

4.2 Datasets

For experiments with text classification, we use three datasets from the GLUE benchmark (Wang et al., 2018) that were previously leveraged by Shelmanov et al. (2021) and Xin et al. (2021) for the same purpose: Microsoft Research Paraphrase Corpus (MRPC) (Dolan and Brockett, 2005), Corpus of Linguistic Acceptability (CoLA) (Warstadt et al., 2019), and Stanford Sentiment Treebank (SST-2) (Socher et al., 2013). Similar to (Shelmanov et al., 2021), we randomly subsample SST-2 to 10% to emulate a low-resource setting.

The experiments with NER were performed on the widely-used CoNLL-2003 task (Tjong Kim Sang and De Meulder, 2003). For this dataset, we also subsample the training part to 10%.

As an out-of-domain dataset for DDPP MC dropout, we use the IMDB binary sentiment classification dataset (Maas et al., 2011). We randomly select 5,000 instances from its test part and use them to select DPP-generated masks.

The dataset statistics are provided in Table 4 in Appendix A.

4.3 Model Choice and Hyperparameter Selection

For experiments, we use two modern Transformers: the pre-trained ELECTRA model (Clark et al., 2020) with 110 million parameters and DeBERTa (He et al., 2021) with 138 million parameters. They achieve higher performance on the GLUE benchmark in comparison with previous models, such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019).

The optimal hyperparameter values for each triple <Dataset, Regularization Type, Spectral Normalization Usage> are presented in Table 6

Method	Reg. Type	UE Score	MRPC		SST-2		CoLA		CoNLL-2003 (token level)		CoNLL-2003 (seq. level)	
			RCC-AUC ↓	RPP ↓	RCC-AUC ↓	RPP ↓	RCC-AUC ↓	RPP ↓	RCC-AUC ↓	RPP ↓	RCC-AUC ↓	RPP ↓
MD	-	MD	13.69±1.25	1.88±0.13	13.08±2.58	0.86±0.15	41.73±1.45	1.96±0.04	10.33±3.55	0.15±0.04	17.05±5.07	2.05±0.45
MD	CER	MD	13.61±1.82	1.87±0.22	14.10±2.69	0.96±0.16	42.50±2.65	2.00±0.07	**6.82±0.90**	**0.10±0.01**	**16.92±2.51**	**1.87±0.23**
MD	metric	MD	13.91±2.35	1.89±0.29	12.03±2.04	0.85±0.15	40.29±2.09	2.02±0.09	10.01±2.56	0.15±0.03	17.67±3.92	2.09±0.36
MD SN (ours)	-	MD	13.44±1.28	1.85±0.20	**11.77±1.33**	**0.83±0.08**	40.07±3.62	1.95±0.16	7.21±1.34	0.11±0.02	17.29±3.58	2.01±0.37
MD SN (ours)	CER	MD	14.41±1.96	1.94±0.21	12.32±1.37	0.85±0.10	**37.82±2.91**	**1.90±0.12**	6.95±1.50	0.11±0.02	17.76±4.00	2.06±0.42
MD SN (ours)	metric	MD	**12.04±1.33**	**1.56±0.12**	12.05±1.42	0.84±0.07	39.37±2.00	1.97±0.15	6.90±1.21	0.11±0.02	17.02±3.39	2.01±0.40
SNGP	-	SNGP	14.52±2.48	2.00±0.35	16.08±4.18	1.02±0.18	51.96±1.89	2.64±0.07	56.43±23.03	0.60±0.22	44.80±11.00	5.06±1.01
SR SN	-	MP	18.83±3.89	2.46±0.46	19.02±6.07	1.21±0.35	81.25±12.56	3.40±0.33	7.46±1.39	0.12±0.02	20.13±3.50	2.30±0.26
SR	CER	MP	14.62±1.62	2.02±0.19	14.56±2.14	1.00±0.14	56.97±9.69	2.53±0.15	6.84±1.41	0.11±0.02	21.31±1.63	2.49±0.25
SR	metric	MP	18.39±2.94	2.40±0.27	16.90±3.12	1.16±0.24	44.54±2.11	2.22±0.15	6.51±1.07	0.10±0.02	20.32±1.68	2.32±0.23
SR (baseline)	-	MP	22.32±8.08	2.58±0.65	17.93±3.84	1.22±0.28	49.48±3.71	2.35±0.25	6.08±0.62	0.10±0.01	18.81±3.35	2.21±0.29

Table 2: Results of deterministic methods with different types of regularization (ELECTRA model). The best results are highlighted with the bold font, the best results for each method are underlined.

in Appendix A. For the optimal hyperparameter search, we split the original training data into training and validation subsets in a ratio of 80 to 20 and apply Bayesian optimization with early stopping. For text classification, we use *accuracy* as an objective metric, and for sequence tagging, we use *span-based F1-score* (Tjong Kim Sang and De Meulder, 2003). Sets of pre-defined values for each hyperparameter are given in the caption of Table 6. After the hyperparameter search is completed, we train the model on the original training set using the optimal values.

The hyperparameters for UE methods are presented in Table 9 in Appendix A. The values for the DDPP MC dropout and MD SN are chosen using a grid search, while validating on the held-out validation dataset with RCC-AUC as an objective. For deep ensemble, we use random subsampling of the training set with a fixed ratio of 90%.

The hardware configuration for experiments is provided in Table 5 in Appendix A.

5 Results and Discussion

5.1 Monte Carlo Dropout and Regularization

The results of methods based on MC dropout and loss regularization are presented in Table 1 (for ELECTRA). The standard computationally intensive MC dropout achieves big improvements over the SR baseline on all text classification datasets and the sequence-level CoNLL-2003 benchmark. For token-level CoNLL-2003, none of the considered methods substantially outperform the baseline. Uncertainty estimation scores BALD and PV have similar results, outperforming SMP on SST-2, while SMP has a slight advantage over them on CoLA and CoNLL-2003.

The DDPP MC dropout method does not outperform the MC dropout. However, DDPP (+DDPP) demonstrates a notable advantage over the SR baseline on text classification datasets SST-2 and CoLA, while both DDPP (+DDPP) and DDPP (+OOD) outperform the baseline on the sequence-level CoNLL-2003 benchmark. The main advantage of the proposed DDPP MC dropout method consists in its much faster inference compared to the computationally expensive standard MC dropout. The DDPP MC dropout has the same computational overhead during inference as the original DPP MC dropout, which is only less than 0.5% of the overhead introduced by the standard MC dropout (Shelmanov et al., 2021).

We conduct an ablation study of the proposed modifications for the original DPP MC dropout. The experimental results of this study presented in Table 12 in Appendix C demonstrate the benefits of using calibration and introducing diversity in mask generation.

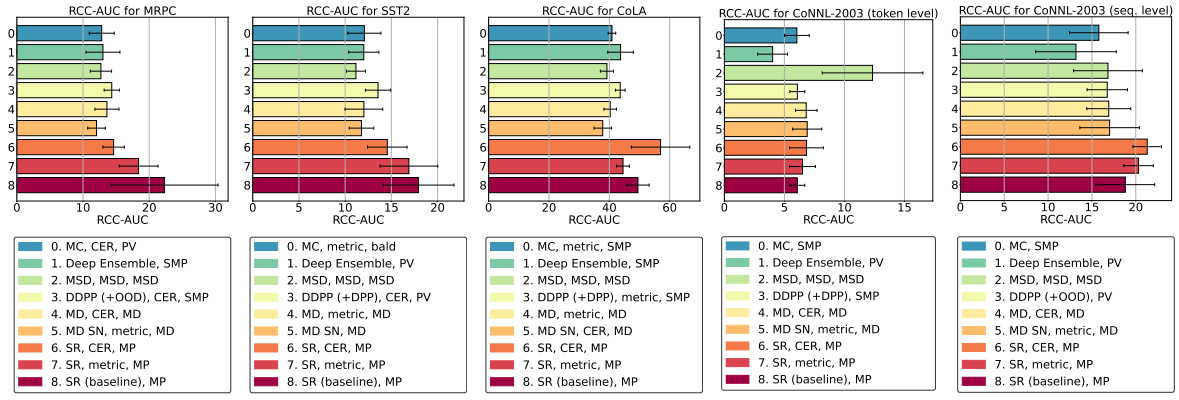
Both metric regularization and CER achieve a substantial advantage over the baseline on text classification datasets SST-2 and MRPC. However, regularization appears to be malignant for NER. Adding loss regularization to MC dropout usually helps to achieve better results on text classification. The best results on SST-2 and CoLA are achieved using metric regularization, while the best result for MRPC is obtained using CER. Regularization and DDPP MC dropout usually complement each other, the results of their combination are slightly better than when they are applied individually for all datasets except CoNLL-2003.

5.2 Deterministic Methods

The results for deterministic methods are presented in Table 2 (for ELECTRA). SNGP gives substantial improvements on the text classification datasets MRPC and SST-2 but significantly falls behind the trivial baseline on CoNLL-2003. The low performance of SNGP for NER can be attributed to the fact that it is initially designed for classification

Method	Reg. Type	UE Score	MRPC		SST-2		CoLA		CoNLL-2003 (token level)		CoNLL-2003 (seq. level)	
			RCC-AUC ↓	RPP ↓	RCC-AUC ↓	RPP ↓	RCC-AUC ↓	RPP ↓	RCC-AUC ↓	RPP ↓	RCC-AUC ↓	RPP ↓
MC	-	SMP	14.38±2.07	1.76±0.19	14.00±2.20	0.91±0.15	42.95±5.98	2.01±0.15	6.04±1.03	0.09±0.02	15.79±3.34	1.80±0.35
MC	CER	PV	12.82±1.89	1.60±0.13	12.18±1.20	0.80±0.10	46.84±9.19	2.11±0.23	6.92±1.22	0.10±0.02	17.05±3.14	1.91±0.36
MC	metric	BALD	14.55±2.31	1.73±0.23	12.08±1.79	0.79±0.10	43.76±0.55	2.08±0.07	6.91±1.02	0.10±0.01	17.47±1.85	1.98±0.30
MC	metric	SMP	13.39±1.19	1.72±0.20	13.55±1.65	0.90±0.14	40.88±1.25	2.01±0.09	6.30±0.98	0.10±0.01	16.81±1.40	1.80±0.23
Deep Ensemble	-	PV	20.70±4.24	2.10±0.35	12.02±1.63	**0.71±0.07**	50.15±5.57	2.21±0.19	**4.02±1.24**	**0.06±0.02**	**13.18±4.60**	**1.54±0.57**
Deep Ensemble	-	SMP	13.01±2.57	1.68±0.27	12.13±1.27	0.79±0.08	43.73±4.25	2.05±0.19	4.16±1.37	**0.06±0.02**	13.93±4.88	1.57±0.58
MSD	MSD	DS	12.70±1.61	1.74±0.25	**11.17±1.03**	0.78±0.06	39.21±2.18	**1.90±0.12**	12.34±4.19	0.18±0.05	16.83±3.92	1.94±0.25
DDPP (+DPP) (ours)	-	PV	22.30±7.15	2.58±0.65	16.70±1.38	1.12±0.12	49.75±3.96	2.44±0.29	6.12±0.71	0.10±0.01	16.78±2.44	1.93±0.20
DDPP (+DPP) (ours)	-	SMP	21.79±7.72	2.57±0.68	17.55±3.03	1.19±0.23	47.86±5.51	2.39±0.31	6.08±0.62	0.10±0.01	17.71±2.77	2.05±0.23
DDPP (+DPP) (ours)	CER	PV	15.12±2.27	2.03±0.24	13.56±1.37	0.91±0.14	54.51±8.80	2.58±0.22	6.98±0.98	0.11±0.02	19.44±1.15	2.13±0.17
DDPP (+DPP) (ours)	CER	SMP	14.75±1.43	2.02±0.16	14.47±1.63	0.99±0.11	54.01±9.79	2.55±0.18	6.91±1.13	0.11±0.02	20.66±1.53	2.31±0.08
DDPP (+DPP) (ours)	metric	SMP	18.45±2.88	2.41±0.26	16.78±3.43	1.14±0.26	43.61±1.61	2.16±0.11	6.92±1.32	0.11±0.02	19.11±2.14	2.16±0.22
DDPP (+OOD) (ours)	-	PV	22.73±7.45	2.65±0.59	19.05±2.95	1.29±0.23	51.11±12.03	2.37±0.34	6.32±0.72	0.10±0.01	16.75±2.31	1.94±0.21
DDPP (+OOD) (ours)	-	SMP	22.31±7.80	2.60±0.65	19.86±3.83	1.36±0.29	50.14±9.73	2.32±0.30	6.09±0.67	0.10±0.01	17.76±2.75	2.06±0.23
DDPP (+OOD) (ours)	CER	BALD	15.03±1.85	2.08±0.24	14.37±2.22	0.96±0.14	57.48±9.37	2.54±0.26	7.41±1.29	0.12±0.02	25.30±3.36	3.00±0.24
DDPP (+OOD) (ours)	CER	SMP	14.34±1.15	1.99±0.16	15.88±1.96	1.08±0.13	59.32±11.86	2.53±0.20	6.88±1.24	0.11±0.02	21.06±1.96	2.35±0.14
DDPP (+OOD) (ours)	metric	SMP	18.55±3.06	2.42±0.27	17.08±3.78	1.14±0.26	43.67±1.77	2.15±0.11	6.71±1.18	0.10±0.02	19.01±2.30	2.16±0.25
MD	CER	MD	13.61±1.82	1.87±0.22	14.10±2.69	0.96±0.16	42.50±2.65	2.00±0.07	6.82±0.90	0.10±0.01	16.92±2.51	1.87±0.23
MD	metric	MD	13.91±2.35	1.89±0.29	12.03±2.04	0.85±0.15	40.29±2.09	2.02±0.09	10.01±2.56	0.15±0.03	17.67±3.92	2.09±0.36
MD SN (ours)	-	MD	13.44±1.28	1.85±0.20	11.77±1.33	0.83±0.08	40.07±3.62	1.95±0.16	7.21±1.34	0.11±0.02	17.29±3.58	2.01±0.37
MD SN (ours)	CER	MD	14.41±1.96	1.94±0.21	12.32±1.37	0.85±0.10	**37.82±2.91**	**1.90±0.12**	6.95±1.50	0.11±0.02	17.76±4.00	2.06±0.42
MD SN (ours)	metric	MD	**12.04±1.33**	**1.56±0.12**	12.05±1.42	0.84±0.07	39.37±2.00	1.97±0.15	6.90±1.21	0.11±0.02	17.02±3.39	2.01±0.40
SR	CER	MP	14.62±1.62	2.02±0.19	14.56±2.14	1.00±0.14	56.97±9.69	2.53±0.15	6.84±1.41	0.11±0.02	21.31±1.63	2.49±0.25
SR	metric	MP	18.39±2.94	2.40±0.27	16.90±3.12	1.16±0.24	44.54±2.11	2.22±0.15	6.51±1.07	0.10±0.02	20.32±1.68	2.32±0.23
SR (baseline)	-	MP	22.32±8.08	2.58±0.65	17.93±3.84	1.22±0.28	49.48±3.71	2.35±0.25	6.08±0.62	0.10±0.01	18.81±3.35	2.21±0.29

Table 3: Comparison of the best results for all methods (ELECTRA model). The computationally intensive methods are at the top of the table; the computationally cheap methods are at the bottom. The best results overall are highlighted with the bold font, the best results for computationally cheap methods are underlined.



a) MRPC b) SST2 c) CoLA d) CoNLL token. level e) CoNLL seq. level

Figure 1: RCC-AUC↓ of the best UE methods for the ELECTRA model.

tasks rather than sequence tagging. MD yields much bigger improvements over the SR baseline on all datasets and significantly outperforms SNGP. MD SN is able to improve the misclassification detection performance even further for MRPC, SST-2, and CoLA.

We also conduct an ablation study (Table 2), in which we use the spectral normalization without MD. We see that SN on its own, as expected, mostly does not improve the UE performance; the results usually are even slightly worse than the baseline.

Regularization also helps to improve the results of methods based on the Mahalanobis distance. For both MD and MD SN, regularization helps on CoLA and CoNLL-2003. For MD, it also helps on SST-2, while for MD SN, regularization improves the results on MRPC. We note that regularization reduces the gap between MD and MD SN on text classification datasets and even gives a slight advantage to MD over MD SN on CoNLL-2003.

The best results across all deterministic methods for text classification datasets are achieved by MD SN. The biggest improvements are obtained on MRPC, where regularized MD SN reduces RCC-AUC by more than 46% compared to the baseline.

5.3 Best Results

Table 3 and Figure 1 compare results of the best methods in each group for ELECTRA. Table 11 and Figure 3 in Appendix B show the best results for DeBERTa. In these tables and figures, we also present the results of deep ensemble (Lakshminarayanan et al., 2017), which is a strong yet computationally intensive baseline (Ashukha et al., 2020), and results of another recently proposed computationally intensive method called MSD (He et al., 2020) that leverage "mix-up" (Thulasidasan et al., 2019), "self-ensembling", MD,

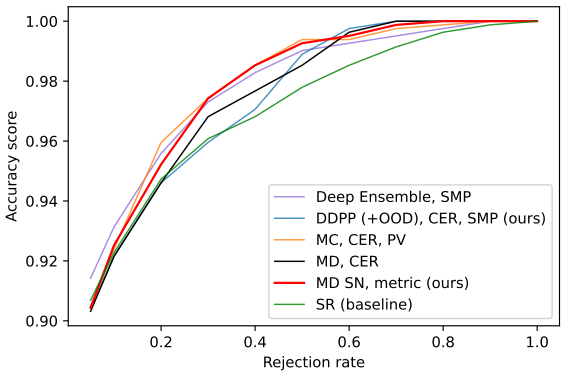


Figure 2: Median values of accuracy rejection curves for selected methods on MRPC (ELECTRA model).

and the MC dropout (all layers are activated).

We can see that it is possible to substantially improve misclassification detection performance and achieve even better results than MC dropout, deep ensemble, or MSD almost with no overhead in terms of memory consumption and amount of computation. For text classification and for both models, computationally cheap methods are either better or on par with the expensive counterparts. However, for NER, we see that the latter methods seriously fall behind deep ensemble and MC dropout. On the token-level CoNLL-2003 benchmark, only deep ensemble substantially outperforms the SR baseline. On the sequence-level CoNLL-2003 benchmark, MD with CER, DDPP (+DDP) PV, and DDPP (+OOD) PV improve upon SR, but only approach the performance of computationally intensive methods.

The proposed in this work MD SN method outperforms all other computationally efficient alternatives on text classification datasets. For both models, it even substantially outperforms all computationally expensive methods on the CoLA dataset, while on other text classification datasets it is on par with them. Another method proposed in this work, DDPP MC dropout, empowered with regularization techniques, is able to substantially reduce the gap between the SR baseline and computationally intensive UE methods, while introducing only a fraction of their overhead.

Figure 2 also presents accuracy rejection curves for selected methods on MRPC. The figure shows that if we reject 20% of instances using UE obtained with MC dropout and ask human experts to label these uncertain objects, the accuracy score of such a human-machine hybrid system will increase from 88.4% to 96.0%, which is 1.3% better than the SR baseline. Such an additional gain over the SR baseline can be crucial for safe-critical applications. Deep ensemble and MD SN are close to each other and achieve 95.6% and 95.2% of accuracy correspondingly. Rejecting 40% of most uncertain instances gives 98.2% of accuracy for the computationally-intensive deep ensemble, while the proposed cheap MD SN method yields even better result with 98.5% of accuracy, which is 1.7% higher than the result of the SR baseline.

6 Conclusion

Our extensive empirical investigation on text classification and NER tasks shows that computationally cheap UE methods are able to substantially improve misclassification detection for Transformers, performing on par or even better than computationally intensive MC dropout and deep ensemble. The proposed in this work method based on the Mahalanobis distance and spectral normalization of a weight matrix (MD SN) achieves the best results among other computationally cheap methods on text classification datasets and is on par with expensive methods. This method does not require seriously modifying a model architecture, extra memory storage, and introduces only a little amount of additional computation during inference.

We also show that our modification of DPP MC dropout that leverages the diversity of generated dropout masks, which is also a computationally cheap method, is able to outperform the softmax response baseline and approach the computationally intensive methods on NER. Finally, we find that regularization can slightly improve the results of methods based on MC dropout and the Mahalanobis distance in text classification.

The spectral normalization is theoretically proven to ensure bi-Lipschitz constraint on the transformation defined by the standard residual connection network (Liu et al., 2020). However, the self-attention blocks in Transformers have a more complicated architecture than the layers of standard ResNets, which means that the theoretical guarantees for them do not hold in general. In future work, we are looking forward to investigating other techniques to ensure bi-Lipschitz constraint on self-attention blocks, which might further improve deterministic methods for uncertainty estimation of Transformers.

Acknowledgements

We thank anonymous reviewers for their insightful suggestions to improve this paper. The work was supported by a grant for research centers in the field of artificial intelligence (agreement identifier 000000D730321P5Q0002 dated November 2, 2021 No. 70-2021-00142 with ISP RAS).

References

Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry P. Vetrov. 2020. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight uncertainty in neural network. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1613–1622. JMLR.org.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Stroudsburg, PA, USA. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Ran El-Yaniv and Yair Wiener. 2010. On the foundations of noise-free selective classification. *J. Mach. Learn. Res.*, 11:1605–1641.

Angelos Filos, Sebastian Farquhar, Aidan N. Gomez, Tim G. J. Rudner, Zachary Kenton, Lewis Smith, Milad Alizadeh, Arnoud de Kroon, and Yarin Gal. 2019. A systematic comparison of bayesian deep learning robustness in diabetic retinopathy tasks. *CoRR*, abs/1912.10481.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA. PMLR.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1183–1192. PMLR.

Yonatan Geifman and Ran El-Yaniv. 2017. Selective classification for deep neural networks. *Advances in Neural Information Processing Systems*, 30:4878–4887.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.

Jianfeng He, Xuchao Zhang, Shuo Lei, Zhiqian Chen, Fanglan Chen, Abdulaziz Alhamadani, Bei Xiao, and Chang-Tien Lu. 2020. Towards more accurate uncertainty estimation in text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 8362–8372. Association for Computational Linguistics.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: decoding-enhanced bert with disentangled attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Neil Houlsby, Ferenc Huszar, Zoubin Ghahramani, and Máté Lengyel. 2011. Bayesian active learning for classification and preference learning. *CoRR*, abs/1112.5745.

Yibo Hu and Latifur Khan. 2021. Uncertainty-aware reliable text classification. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pages 628–636. ACM.

Siddhartha Jain, Ge Liu, Jonas Mueller, and David Gifford. 2020. Maximizing overall diversity for improved uncertainty estimates in deep ensembles. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):4264–4271.

Alex Kulesza and Ben Taskar. 2012. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NeurIPS'17, page 6405–6416, Red Hook, NY, USA. Curran Associates Inc.

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 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, volume 31, pages 7167–7177.

Jeremiah Z. Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax-Weiss, and Balaji Lakshminarayanan. 2020. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Andrey Malinin and Mark J. F. Gales. 2018. Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7047–7058.

Jishnu Mukhoti, Andreas Kirsch, Joost van Amersfoort, Philip H. S. Torr, and Yarin Gal. 2021. Deterministic neural networks with appropriate inductive biases capture epistemic and aleatoric uncertainty. *CoRR*, abs/2102.11582.

Alexander Podolskiy, Dmitry Lipin, Andrey Bout, Ekaterina Artemova, and Irina Piontkovskaya. 2021. Revisiting mahalanobis distance for transformer-based out-of-domain detection. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13675–13682. AAAI Press.

Carl Edward Rasmussen. 2003. Gaussian processes in machine learning. In *Advanced Lectures on Machine Learning, ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures*, volume 3176 of *Lecture Notes in Computer Science*, pages 63–71. Springer.

Burr Settles. 2009. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Artem Shelmanov, Evgenii Tsymbalov, Dmitri Puzyrev, Kirill Fedyanin, Alexander Panchenko, and Maxim Panov. 2021. How certain is your Transformer? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1833–1840, Online. Association for Computational Linguistics.

Yilin Shen, Yen-Chang Hsu, Avik Ray, and Hongxia Jin. 2021. Enhancing the generalization for intent classification and out-of-domain detection in SLU. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2443–2453, Online. Association for Computational Linguistics.

Lewis Smith and Yarin Gal. 2018. Understanding measures of uncertainty for adversarial example detection. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 560–569. AUAI Press.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Sunil Thulasidasan, Gopinath Chennupati, Jeff A. Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. 2019. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13888–13899.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural*

Language Learning at HLT-NAACL 2003, pages 142–147.

Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. 2020. Uncertainty estimation using a single deep deterministic neural network. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9690–9700. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Ji Xin, Raphael Tang, Yaoliang Yu, and Jimmy Lin. 2021. The art of abstention: Selective prediction and error regularization for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1040–1051, Online. Association for Computational Linguistics.

Zhiyuan Zeng, Keqing He, Yuanmeng Yan, Zijun Liu, Yanan Wu, Hong Xu, Huixing Jiang, and Weiran Xu. 2021. Modeling discriminative representations for out-of-domain detection with supervised contrastive learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 870–878, Online. Association for Computational Linguistics.

Xuchao Zhang, Fanglan Chen, Chang-Tien Lu, and Naren Ramakrishnan. 2019. Mitigating uncertainty in document classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3126–3136, Minneapolis, Minnesota. Association for Computational Linguistics.

A Dataset Statistics, Hyperparameter Values, and Hardware Configuration

Datasets	Train	Test	# Labels
MRPC	3.7K	0.4K	2
CoLA	8.6K	1.0K	2
SST-2	67.3K	0.9K	2
CoNLL-2003	14.0K/203.6K	3.5K/46.4K	9

Table 4: Dataset statistics. The table presents the number of sequences for the training and test parts of the datasets. For CoNLL-2003, the table presents both the number of sequences and tokens because for NER, we evaluate both sequence-level and token-level UE scores. For the datasets from the GLUE benchmark (MRPC, CoLA, SST-2), we used the available validation set as the test set.

CPU	2 Intel Xeon Platinum 8168, 2.7 GHz
CPU Cores	24
GPU	NVIDIA Tesla v100 GPU
GPU Memory	32 GB

Table 5: Hardware configuration used in experiments.

Dataset	Reg. Type	Spect. Norm.	Objective Score	Reg. Lambda	Margin	Learning Rate	Num. Epochs	Batch Size	Weight Decay
CoLA	–	1.0	0.876	-	-	3e-5	15	32	1e-1
CoLA	–	-	0.88	-	-	1e-5	8	4	1e-1
CoLA	CER	0.4	0.88	1.0	-	3e-5	11	32	1e-1
CoLA	CER	-	0.882	1e-2	-	9e-6	7	4	1e-2
CoLA	Metric	0.4	0.868	1e-2/1.0	0.1	3e-5	11	32	1e-1
CoLA	Metric	-	0.878	1e-2/2e-2	0.25	9e-6	12	4	1e-1
CoLA	MSD	-	0.877	1e-1/6e-3	0.55	3e-5	7	64	1e-2
MRPC	–	1.0	0.858	-	-	3e-5	11	32	1e-1
MRPC	–	-	0.867	-	-	5e-5	12	32	1e-1
MRPC	CER	3.0	0.871	1.0	-	3e-5	12	4	0
MRPC	CER	-	0.871	2e-1	-	5e-5	7	16	1e-2
MRPC	Metric	0.4	0.845	2e-3/1e-1	0.01	3e-5	10	32	0
MRPC	Metric	-	0.844	1e-2/1.0	0.1	3e-5	11	32	1e-1
MRPC	MSD	-	0.871	1e-1/6e-3	0.5	3e-5	11	8	1e-2
SST-2	–	0.8	0.939	-	-	5e-5	7	64	1e-2
SST-2	–	-	0.936	-	-	1e-5	15	64	1e-1
SST-2	CER	0.8	0.938	1.0	-	3e-5	14	16	1e-1
SST-2	CER	-	0.938	2e-2	-	3e-5	5	64	0
SST-2	Metric	2.0	0.939	8e-3/2e-2	10.0	3e-5	5	64	0
SST-2	Metric	-	0.941	8e-3/2e-2	10.0	3e-5	5	64	0
SST-2	MSD	-	0.942	1e-1/6e-3	0.55	3e-5	7	64	1e-2
CoNLL-2003	–	3.0	0.922	-	-	5e-5	13	8	1e-2
CoNLL-2003	–	-	0.909	-	-	5e-5	6	8	1e-2
CoNLL-2003	CER	1.0	0.913	1e-1	-	5e-5	13	8	1e-2
CoNLL-2003	CER	-	0.912	2e-3	-	2e-5	15	16	1e-2
CoNLL-2003	Metric	3.0	0.911	6e-3/1e-3	0.05	5e-5	15	8	0
CoNLL-2003	Metric	-	0.909	1e-3/1e-1	0.025	5e-5	13	8	1e-2
CoNLL-2003	MSD	-	0.928	1.0/5e-3	0.95	5e-5	9	8	0

Table 6: Optimal hyperparameters for the experiments with ELECTRA except SNGP. "Objective score" refers to the accuracy score for classification / F1-score for sequence tagging on the validation sample. For the metric regularization the reg. lambda column contains λ and $\varepsilon\lambda$ parameters. For the MSD method, the reg. lambda column contains λ_1 and λ_2 parameters and the margin column contains Ω parameter. We select hyperparameter values from the following pre-defined list:
Reg. lambda (λ) (and also (λ_1) and (λ_2)): [1e-3, 2e-3, 3e-3, 5e-3, 6e-3, 8e-3, 1e-2, 2e-2, 5e-2, 1e-1, 2e-1, 1];
Reg. lambda (for metric regularization): [1e-2, 2.5e-2, 5e-2, 1e-1, 2.5e-1, 5e-1, 1.0, 2.5, 5.0, 10.0];
Reg. $\varepsilon\lambda$: [1e-3, 2e-3, 3e-3, 5e-3, 6e-3, 8e-3, 1e-2, 2e-2, 5e-2, 1e-1, 2e-1, 1];
Margin (γ): [1e-2, 2.5e-2, 5e-2, 1e-1, 2.5e-1, 5e-1, 1.0, 2.5, 5.0, 10.0];
Omega (Ω): [0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 1.0];
Spect. Norm.: [0.4, 0.6, 0.8, 1.0, 2.0, 3.0];
Learning rate: [5e-6, 6e-6, 7e-6, 9e-6, 1e-5, 2e-5, 3e-5, 5e-5, 7e-5, 1e-4];
Num. of epochs: $\{n \in \mathbb{N} | 2 \leq n \leq 15\}$;
Batch size: [4, 8, 16, 32, 64];
Weight decay: [0, 1e-2, 1e-1].

Dataset	Objective Score	Learning Rate	Num. Epochs	Batch Size	Weight Decay
CoLA	0.879	6e-6	10	16	0
MRPC	0.867	3e-5	9	16	0
SST-2	0.917	2e-5	6	8	1e-1
CoNLL-2003	0.887	3e-5	85	4	1e-1

Table 7: Optimal hyperparameters for the experiments with SNGP and ELECTRA. For text classification datasets, we use the same pre-defined list of possible values for each hyperparameter. For CoNLL-2003, the following value ranges are used for the *number of epochs* and *learning rate*:
Learning rate: [9e-6, 1e-5, 2e-5, 3e-5, 5e-5, 7e-5, 1e-4];
Num. of epochs: $\{n \in \mathbb{N} | 10 \leq n \leq 100\}$.

Dataset	Reg. Type	Spect. Norm.	Objective Score	Reg. Lambda	Margin	Learning Rate	Num. Epochs	Batch Size	Weight Decay
CoLA	–	0.4	0.854	-	-	7e-6	8	4	1e-1
CoLA	–	-	0.86	-	-	7e-6	13	4	0
CoLA	CER	0.4	0.857	5e-2	-	2e-5	11	32	1e-2
CoLA	CER	-	0.854	2e-1	-	9e-6	15	16	1e-2
CoLA	Metric	0.8	0.86	1.0/3e-3	0.1	6e-6	11	4	1e-2
CoLA	Metric	-	0.862	8e-3/6e-3	0.025	1e-5	12	4	1e-1
CoLA	MSD	-	0.857	5e-2/3e-3	0.65	3e-5	12	32	0
MRPC	–	0.8	0.879	-	-	9e-6	11	16	1e-1
MRPC	–	-	0.889	-	-	3e-5	12	4	1e-1
MRPC	CER	0.6	0.88	6e-3	-	2e-5	15	16	0
MRPC	CER	-	0.88	1.0	-	2e-5	10	16	1e-2
MRPC	Metric	1.0	0.883	8e-3/5e-2	2.5	2e-5	14	16	1e-1
MRPC	Metric	-	0.885	6e-3/1.0	5.0	9e-6	13	8	1e-1
MRPC	MSD	-	0.876	1e-2/2e-2	0.5	9e-6	12	8	1e-1
SST-2	–	0.6	0.901	-	-	9e-6	11	8	1e-1
SST-2	–	-	0.906	-	-	3e-5	5	16	1e-2
SST-2	CER	0.6	0.902	1.0	-	7e-6	12	16	1e-2
SST-2	CER	-	0.902	1e-1	-	6e-6	6	4	0
SST-2	Metric	0.6	0.902	6e-3/5e-3	0.025	5e-5	6	64	1e-1
SST-2	Metric	-	0.902	6e-3/8e-3	0.05	7e-6	8	16	1e-1
SST-2	MSD	-	0.929	1e-2/3e-3	0.95	1e-5	11	4	1e-2
CoNLL-2003	–	1.0	0.897	-	-	5e-5	3	4	1e-2
CoNLL-2003	–	-	0.902	-	-	5e-5	12	32	0
CoNLL-2003	CER	1.0	0.901	5e-2	-	1e-4	10	16	0
CoNLL-2003	CER	-	0.899	2e-1	-	2e-5	13	4	1e-1
CoNLL-2003	Metric	2.0	0.898	2e-3/5e-3	5.0	7e-5	12	8	1e-2
CoNLL-2003	Metric	-	0.908	2e-2/1e-1	0.5	3e-5	14	8	1e-2
CoNLL-2003	MSD	-	0.935	1e-1/5e-3	0.95	3e-5	15	4	1e-1

Table 8: Optimal hyperparameters for all the experiments with DeBERTa except SNGP. "Objective score" refers to the accuracy score for classification / F1-score for sequence tagging on the validation sample. For the metric regularization the reg. lambda column contains λ and $\varepsilon\lambda$ parameters. For the MSD method, the reg. lambda column contains λ_1 and λ_2 parameters and the margin column contains Ω parameter. We select hyperparameter values from the following pre-defined list:
Reg. lambda (λ) (and also (λ_1) and (λ_2)): [1e-3, 2e-3, 3e-3, 5e-3, 6e-3, 8e-3, 1e-2, 2e-2, 5e-2, 1e-1, 2e-1, 1];
Reg. lambda (for metric regularization): [1e-2, 2.5e-2, 5e-2, 1e-1, 2.5e-1, 5e-1, 1.0, 2.5, 5.0, 10.0];
Reg. $\varepsilon\lambda$: [1e-3, 2e-3, 3e-3, 5e-3, 6e-3, 8e-3, 1e-2, 2e-2, 5e-2, 1e-1, 2e-1, 1];
Margin (γ): [1e-2, 2.5e-2, 5e-2, 1e-1, 2.5e-1, 5e-1, 1.0, 2.5, 5.0, 10.0];
Omega (Ω): [0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 1.0];
Learning rate: [5e-6, 6e-6, 7e-6, 9e-6, 1e-5, 2e-5, 3e-5, 5e-5, 7e-5, 1e-4];
Num. of epochs: $\{n \in \mathbb{N} | 2 \leq n \leq 15\}$;
Batch size: [4, 8, 16, 32, 64];
Weight decay: [0, 1e-2, 1e-1].

Dataset	Method	Dropout Ratio	Committee Size	Max. Frac.	Kernel Type	DDPP Mask Pool Size	DDPP Kernel Type
CoLA	DDPP (+OOD)	-	50	0.45	corr.	100	-
CoLA	DDPP (+DPP)	-	50	0.4	corr.	100	RBF
CoLA	MC dropout	0.1	20	-	-	-	-
CoLA	Deep Ensemble	-	5	-	-	-	-
MRPC	DDPP (+OOD)	-	50	0.4	corr.	100	-
MRPC	DDPP (+DPP)	-	50	0.55	corr.	100	RBF
MRPC	MC dropout	0.1	20	-	-	-	-
MRPC	Deep Ensemble	-	5	-	-	-	-
SST-2	DDPP (+OOD)	-	50	0.35	corr.	100	-
SST-2	DDPP (+DPP)	-	50	0.45	corr.	100	RBF
SST-2	MC dropout	0.1	20	-	-	-	-
SST-2	Deep Ensemble	-	5	-	-	-	-
CoNLL-2003	DDPP (+OOD)	-	20	0.6	corr.	100	-
CoNLL-2003	DDPP (+DPP)	-	20	0.6	corr.	100	RBF
CoNLL-2003	MC dropout	0.1	20	-	-	-	-
CoNLL-2003	Deep Ensemble	-	5	-	-	-	-

Table 9: Optimal hyperparameter values for UE methods based on MC dropout and deep ensemble with the ELECTRA model. These parameters denote the following:
Dropout Ratio – probability of a neuron to be zeroed during inference in a dropout layer;
Committee Size – a number of ensemble elements or stochastic forward passes in the MC dropout;
Max. Frac. – a maximum number of active neurons in a DPP mask;
Kernel Type – type of a kernel in a DPP mask;
DDPP Mask Pool Size – a number of masks in a pool, from which DDPP selects a diverse set of masks;
DDPP Kernel Type – a type of a kernel for a DDPP mask.

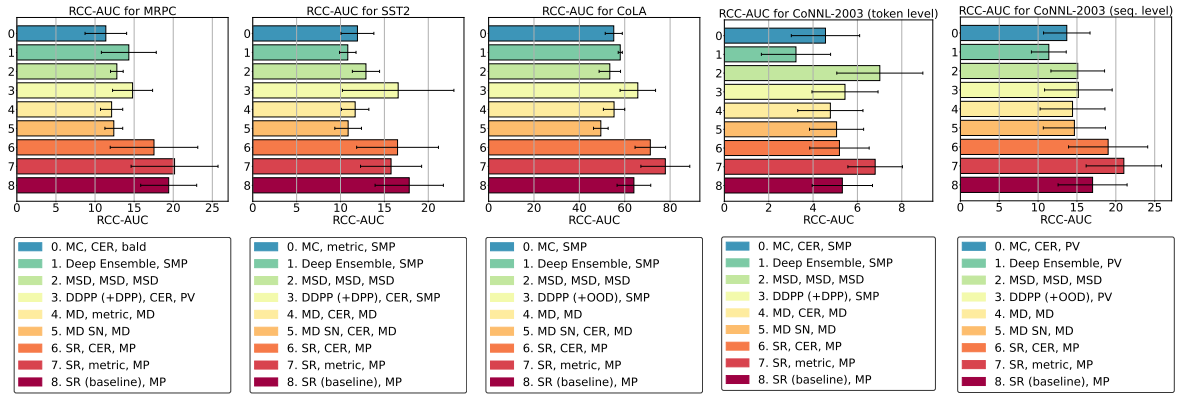
Dataset	Method	Dropout Ratio	Committee Size	Max. Frac.	Kernel Type	DDPP Mask Pool Size	DDPP Kernel Type
CoLA	DDPP (+OOD)	-	50	0.45	corr.	100	-
CoLA	DDPP (+DPP)	-	50	0.6	corr.	100	RBF
CoLA	MC dropout	0.1	20	-	-	-	-
CoLA	Deep Ensemble	-	5	-	-	-	-
MRPC	DDPP (+OOD)	-	50	0.45	corr.	100	-
MRPC	DDPP (+DPP)	-	50	0.6	corr.	100	RBF
MRPC	MC dropout	0.1	20	-	-	-	-
MRPC	Deep Ensemble	-	5	-	-	-	-
SST-2	DDPP (+OOD)	-	50	0.45	corr.	100	-
SST-2	DDPP (+DPP)	-	50	0.6	corr.	100	RBF
SST-2	MC dropout	0.1	20	-	-	-	-
SST-2	Deep Ensemble	-	5	-	-	-	-
CoNLL-2003	DDPP (+OOD)	-	20	0.45	corr.	100	-
CoNLL-2003	DDPP (+DPP)	-	20	0.3	corr.	100	RBF
CoNLL-2003	MC dropout	0.1	20	-	-	-	-
CoNLL-2003	Deep Ensemble	-	5	-	-	-	-

Table 10: Optimal hyperparameter values for UE methods based on MC dropout and deep ensemble with the DeBERTa model. These parameters denote the following:
Dropout Ratio – probability of a neuron to be zeroed during inference in a dropout layer;
Committee Size – a number of ensemble elements or stochastic forward passes in the MC dropout;
Max. Frac. – a maximum number of active neurons in a DPP mask;
Kernel Type – type of a kernel in a DPP mask;
DDPP Mask Pool Size – a number of masks in a pool, from which DDPP selects a diverse set of masks;
DDPP Kernel Type – a type of a kernel for a DDPP mask.

B Additional Experimental Results with DeBERTa

Method	Reg. Type	UE Score	MRPC		SST-2		CoLA		CoNLL-2003 (token level)		CoNLL-2003 (seq. level)	
			RCC-AUC ↓	RPP ↓	RCC-AUC ↓	RPP ↓	RCC-AUC ↓	RPP ↓	RCC-AUC ↓	RPP ↓	RCC-AUC ↓	RPP ↓
MC	-	SMP	15.06±3.93	1.85±0.40	13.59±2.30	0.91±0.14	55.17±3.76	2.62±0.16	4.91±0.95	0.07±0.01	14.29±3.50	1.74±0.37
MC	CER	PV	11.53±2.73	1.42±0.24	12.75±3.89	0.85±0.19	56.65±3.27	2.65±0.14	5.10±1.73	0.07±0.02	13.69±2.99	1.79±0.39
MC	CER	BALD	**11.38±2.66**	**1.42±0.23**	12.90±4.15	0.86±0.19	57.62±3.88	2.68±0.15	5.21±1.67	0.08±0.02	14.21±2.94	1.82±0.36
MC	CER	SMP	12.30±3.19	1.48±0.27	12.26±3.04	0.85±0.17	55.28±3.20	2.59±0.13	4.56±1.54	0.07±0.02	13.91±3.08	1.79±0.39
MC	metric	SMP	15.18±4.58	1.72±0.31	11.93±1.87	0.81±0.13	62.89±7.57	2.75±0.21	5.91±1.24	0.09±0.02	15.05±3.64	1.80±0.31
Deep Ensemble	-	PV	18.81±5.69	2.01±0.40	12.19±2.31	0.71±0.05	64.80±2.71	2.80±0.10	3.76±1.71	**0.05±0.02**	**11.38±2.24**	1.40±0.33
Deep Ensemble	-	SMP	14.32±3.51	1.77±0.30	**10.83±0.94**	**0.70±0.04**	58.03±0.90	2.70±0.07	**3.23±1.56**	0.05±0.03	11.77±2.42	**1.40±0.32**
MSD	MSD	DS	12.79±0.81	1.80±0.08	12.90±1.55	0.90±0.08	53.43±4.72	2.60±0.20	7.01±1.94	0.10±0.02	15.10±3.45	1.84±0.36
DDPP (+DPP) (ours)	-	PV	18.12±2.53	2.36±0.27	18.41±3.57	1.20±0.19	69.81±7.82	3.40±0.29	5.56±1.51	0.09±0.02	15.63±4.97	1.90±0.52
DDPP (+DPP) (ours)	-	SMP	18.13±3.27	2.30±0.34	17.74±4.17	1.17±0.24	68.12±6.34	3.29±0.23	5.44±1.49	0.08±0.02	17.56±4.97	2.15±0.54
DDPP (+DPP) (ours)	CER	PV	14.80±2.56	1.88±0.22	17.61±7.41	1.10±0.32	73.34±8.08	3.39±0.39	8.15±3.45	0.12±0.03	19.05±4.16	2.48±0.51
DDPP (+DPP) (ours)	CER	SMP	16.69±5.35	1.99±0.45	16.57±6.35	1.08±0.31	72.15±7.10	3.29±0.34	6.18±1.78	0.10±0.02	20.66±5.06	2.69±0.61
DDPP (+OOD) (ours)	-	PV	19.64±5.28	2.45±0.52	17.98±3.12	1.20±0.21	68.49±7.77	3.28±0.32	5.87±1.48	0.09±0.02	15.18±4.35	1.86±0.47
DDPP (+OOD) (ours)	-	SMP	18.86±3.04	2.37±0.36	18.52±3.49	1.23±0.23	65.77±7.82	3.13±0.35	5.45±1.38	0.08±0.02	17.25±4.60	2.09±0.50
DDPP (+OOD) (ours)	CER	BALD	15.59±2.41	2.07±0.30	18.44±4.44	1.23±0.25	71.75±8.22	3.23±0.36	6.45±1.77	0.10±0.02	22.64±5.74	2.90±0.66
DDPP (+OOD) (ours)	metric	BALD	18.96±3.24	2.30±0.26	17.41±4.85	1.14±0.31	94.05±24.27	4.30±0.75	7.69±3.18	0.10±0.02	21.42±2.41	2.57±0.16
MD	-	MD	14.66±3.65	1.98±0.40	12.51±1.97	0.86±0.13	55.30±4.70	2.68±0.19	4.83±1.45	0.07±0.01	14.43±4.17	1.75±0.45
MD	CER	MD	13.48±1.24	1.88±0.19	11.67±1.56	0.85±0.11	57.78±3.86	2.73±0.15	4.78±1.47	0.07±0.02	14.69±4.07	1.87±0.48
MD	metric	MD	12.12±1.42	1.64±0.17	11.81±1.84	0.85±0.13	57.35±4.35	2.74±0.20	5.42±1.28	0.08±0.02	14.51±3.96	1.70±0.30
MD SN (ours)	-	MD	12.40±1.14	1.78±0.18	11.10±1.03	0.78±0.09	52.49±1.44	2.42±0.09	5.06±1.22	0.08±0.01	14.67±4.00	1.79±0.19
MD SN (ours)	CER	MD	13.03±1.49	1.86±0.18	10.87±1.52	0.80±0.11	**49.47±3.23**	**2.36±0.19**	5.92±1.18	0.09±0.01	16.57±3.26	1.97±0.30
SR	CER	MP	17.54±5.60	2.10±0.41	16.50±4.66	1.11±0.26	71.28±6.73	3.22±0.30	5.19±1.34	0.08±0.02	19.01±5.09	2.44±0.64
SR	metric	MP	20.17±5.56	2.38±0.49	15.76±3.48	1.07±0.27	77.90±10.78	3.33±0.48	6.80±1.23	0.11±0.02	21.03±4.85	2.68±0.57
SR (baseline)	-	MP	19.42±3.58	2.44±0.33	17.83±3.89	1.18±0.23	64.05±7.42	3.05±0.30	5.32±1.36	0.08±0.01	17.01±4.44	2.06±0.39

Table 11: Comparison of the best results for all methods (DeBERTa model). The computationally intensive methods are at the top of the table; the computationally cheap methods are at the bottom. The best results overall are highlighted with the bold font, the best results for computationally cheap methods are underlined.



a) MRPC b) SST2 c) CoLA d) CoNLL token level e) CoNLL seq. level

Figure 3: RCC-AUC↓ of the best UE methods for the DeBERTa model.

C Additional Ablation Studies for DDPP

Method	Reg. Type	UE Score	MRPC		SST-2		CoLA		CoNLL-2003 (token level)		CoNLL-2003 (seq. level)	
			RCC-AUC ↓	RPP ↓	RCC-AUC ↓	RPP ↓	RCC-AUC ↓	RPP ↓	RCC-AUC ↓	RPP ↓	RCC-AUC ↓	RPP ↓
DDPP (+DPP) (ours)	-	PV	22.30±7.15	2.58±0.65	16.70±1.38	1.12±0.12	49.75±3.96	2.44±0.29	6.12±0.71	0.10±0.01	16.78±2.44	1.93±0.20
DDPP (+DPP) (ours)	-	BALD	23.08±7.00	2.63±0.63	**16.08±2.37**	**1.05±0.18**	49.59±5.40	2.48±0.31	6.39±0.64	0.10±0.01	21.53±4.77	2.63±0.45
DDPP (+DPP) (ours)	-	SMP	**21.79±7.72**	**2.57±0.68**	17.55±3.03	1.19±0.23	**47.86±5.51**	2.39±0.31	**6.08±0.62**	0.10±0.01	17.71±2.77	2.05±0.23
DDPP (+OOD) (ours)	-	PV	22.73±7.45	2.65±0.59	19.05±2.95	1.29±0.23	51.11±12.03	2.37±0.34	6.32±0.72	0.10±0.01	16.75±2.31	1.94±0.21
DDPP (+OOD) (ours)	-	BALD	23.85±8.39	2.69±0.58	18.27±3.05	1.22±0.23	52.59±12.08	2.42±0.34	6.59±0.69	0.11±0.01	20.56±3.09	2.50±0.26
DDPP (+OOD) (ours)	-	SMP	22.31±7.80	2.60±0.65	19.86±3.83	1.36±0.29	50.14±9.73	**2.32±0.30**	6.09±0.67	0.10±0.01	17.76±2.75	2.06±0.23
DPP	-	PV	23.96±9.77	2.63±0.60	18.60±3.59	1.20±0.23	53.49±4.30	2.43±0.26	6.31±0.56	0.10±0.01	**16.23±2.23**	**1.87±0.21**
DPP	-	BALD	24.94±10.22	2.68±0.58	19.39±4.99	1.21±0.31	54.59±4.09	2.49±0.26	6.49±0.56	0.10±0.01	19.09±3.59	2.27±0.32
DPP	-	SMP	21.83±7.92	2.59±0.65	18.19±3.44	1.23±0.25	51.06±4.51	2.40±0.28	6.18±0.54	**0.10±0.00**	17.28±2.53	1.98±0.21

Table 12: The comparison of original DPP MC dropout with its two modifications DDPP MC dropout (ELECTRA model).