

Received March 4, 2022, accepted April 14, 2022, date of publication April 21, 2022, date of current version April 28, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3168734

Benchmarking Scalable Predictive Uncertainty in Text Classification

JORDY VAN LANDEGHEM^{1,2}, MATTHEW BLASCHKO³, BERTRAND ANCKAERT²,
AND MARIE-FRANCINE MOENS¹

¹LIIR, Department of Computer Science, KU Leuven, 3000 Leuven, Belgium

²Contract.fit, 9800 Deinze, Belgium

³Department of Electrical Engineering, ESAT-PSI, KU Leuven, 3000 Leuven, Belgium

Corresponding author: Jordy Van Landeghem (jordy.vanlandeghem@cs.kuleuven.be)

This work was supported by the Flemish Innovation and Entrepreneurship (VLAIO) through the Baekeland Ph.D. Mandate under Grant HBC.2019.2604.

ABSTRACT This paper explores the question of how predictive uncertainty methods perform in practice in Natural Language Processing, specifically multi-class and multi-label text classification. We conduct benchmarking experiments with 1-D convolutional neural networks and pre-trained transformers on six real-world text classification datasets in which we empirically investigate why popular scalable uncertainty estimation strategies (*Monte-Carlo Dropout*, *Deep Ensemble*) and notable extensions (*Heteroscedastic*, *Concrete Dropout*) underestimate uncertainty. We motivate that uncertainty estimation benefits from combining posterior approximation procedures, linking it to recent research on how ensembles and variational Bayesian methods navigate the loss landscape. We find that our proposed method combination of *Deep Ensemble* with *Concrete Dropout*, by analysis of in-domain calibration, cross-domain classification, and novel class robustness, demonstrates superior performance, even at a smaller ensemble size. Our results corroborate the importance of fine-tuning dropout rate to the text classification task at hand, which individually and as an ensemble impacts model robustness. We observe in ablation that pre-trained transformers severely underperform in novelty detection, limiting the applicability of transfer learning when distribution shift from novel classes can be expected.

INDEX TERMS Bayesian deep learning, natural language processing, text classification, out-of-distribution detection, cross-domain classification.

I. INTRODUCTION

Reliable uncertainty quantification is indispensable for any machine learning system trusted in decision-making in many application domains such as medical diagnosis, self-driving cars and automated document processing. In any typical industrial application, we desire predictive uncertainty to communicate on the model's lack of in-domain knowledge, due to either training data scarcity or model design errors, or its ability to flag potentially noisy, shifted or unknown input data (see [1] for more detail on sources of uncertainty).

Supervised Deep Learning (DL) algorithms have been found to provide “catastrophically overconfident predictions” [2] under data distribution shift. Specifically, novel class distributions can emerge at inference time [3], which

desirably should be detectable in a model's uncertainty. To this end, scalable Bayesian DL (BDL) methods for uncertainty estimation have been recently developed, generating increased interest from practitioners in need of practical solutions. BDL comprises an increasingly large range of theoretically well-motivated predictive uncertainty methods, yet only some are able to scale in network architecture and dataset size. Additionally, most surveys and research output on predictive uncertainty is based on multi-class image classification or regression experiments. We argue that predictive uncertainty methods and how well they scale in Natural Language Processing (NLP), for text classification tasks, is still an under-explored question.

The context of our study is a production-level text classification system for automatically handling incoming communications in information-intensive industries (e.g. legal, banking, insurance). Imagine a digital-first company where

The associate editor coordinating the review of this manuscript and approving it for publication was Nikhil Padhi¹.

each department has its own document classifier operating under a closed world assumption. However, whenever a client mistakenly sends a document (car purchase invoice requesting a loan) to the wrong department (say underwriting or medical claims), this can generate high-confidence false positives that trigger the wrong action (insurance or claim settlement instead of loan application). Similarly, if an insurance broker suddenly decides to completely change the document template that clients use to apply for a car loan, the production model might not find previously salient features which it had learned to rely on for accurate classification. This shows that detection of anomalous inputs and shifting distributions is critical to keep errors in automation low.

We investigate different techniques and procedures for incorporating uncertainty into Deep Learning models for text classification, analyzing the degree to which they can reliably capture uncertainty under extrapolation (outside the support of the training set), both individually and combined in an ensemble. Our findings for individual predictive uncertainty methods are overall consistent with benchmarks in other modalities, with Deep Ensemble reporting greater robustness than approximate Bayesian methods. However, we discover from empirical findings that our newly proposed combinations, particularly *MC Concrete Dropout Ensemble*, can push the bounds by exploiting the in-domain calibration effect of Concrete Dropout and all-round ensemble qualities for increased out-of-domain and novel class robustness.

We intend our work to be used as a survey and benchmark of scalable BDL methods, where the architectures and datasets are drawn from NLP, thereby covering a void in the literature on uncertainty estimation in this field. Next to proposing a well-motivated evaluation methodology, this paper also provides a convenient entry point for practitioners.¹

Our key contributions can be summarized as follows:

- We conduct a benchmarking study of established uncertainty estimation methods applied on real-world text classification datasets. Our analysis focuses on model robustness and uncertainty quality in realistic data distributions. We propose a practical methodology to test the above, resulting in a better understanding of the individual shortcomings of predictive uncertainty methods.
- We motivate and introduce novel combinations of predictive uncertainty methods, providing empirical evidence for their complementary benefits. Through statistical analyses and ablation experiments we discern the importance of certain prior, model or hyperparameter influences on the reliability of predictive uncertainty.

Organization The paper is organized as follows. Subsection I-A overviews related work in uncertainty benchmarking, distribution shift, and uncertainty estimation in NLP. We present core concepts of BDL in Section II to build up a thorough understanding of predictive uncertainty in theory

and practice. We include this introductory text for readers less familiar with uncertainty methods. Subsection II-E critically analyzes the practice of evaluating uncertainty under distribution shift. Subsections II-D and III-A stand central in our work, connecting recent research on how neural networks navigate the loss landscape with posterior approximation procedures, followed by our work's hypotheses on complementary benefits between predictive uncertainty methods.

Section III details our methodological setup from datasets, model architectures, uncertainty estimation and evaluation, to experimental settings. We present in Section IV the results of 3 large benchmarking experiments, followed by 4 smaller ablation studies on important hyperparameters. After closing the discussion in Section V with take-home messages targeting researchers and practitioners interested in uncertainty prediction in text classification, Section VI draws up some limitations of our research. Finally, we synthesize our contributions in Section VII and propose directions for future work on uncertainty research in NLP.

The Appendices support the main text by detailing implementation (A), practical considerations (compute, timings) (B), extended experiments with alternative uncertainty methods (C), and detailed evaluation data for full transparency (D).

A. RELATED WORK

In this Subsection, we overview recent literature on benchmarking the quality of uncertainty quantification in DL and more specifically research on uncertainty estimation for NLP tasks.

Increasingly, there are efforts from the research community to help BDL methods scale to real-world scenarios [4]. Benchmarks are an important tool to help researchers prioritize the right approaches and to inform practitioners which methods are suited for their applications [5]. There is a growing demand for benchmarking in BDL, since methods must be scored both for task performance and uncertainty quality [6], [7]. Rigorously evaluating the latter is considerably more difficult, since depending on the problem setting no direct uncertainty ground-truth exists, requiring a well-defined experimental setup [8].

A standard benchmark in BDL is *UCI* [9], a set of curated regression datasets, which allows to judge uncertainty quality with the predictive log-likelihood metric. However, its general applicability and validity has been criticized on multiple accounts [8], [10], [11].

More recently, [11]–[15] presented large-scale evaluation studies of BDL methods with benchmarking on real-world datasets. These studies motivate data retention and distribution shift as generic protocols for evaluating predictive uncertainty. Similarly, we argue that even mild shifts of data are unavoidable in real-world applications and, conditional to specific distribution shift assumptions (see Subsection II-E), this provides a good testing ground for uncertainty evaluation.

Reference [12] consider two types of distribution shift: (a) *out-of-distribution* (OOD) data from separate datasets,

¹Our benchmarking software [TensorFlow 2] is available at <https://github.com/Jordy-VL/uncertainty-bench>

and (b) *adversarial shift*, where the test distribution consists of perturbed or corrupted ground truth data isolated from training.

In our work we propose novel class detection as an alternative to a), which we motivate to be a more representative experimental setup for testing uncertainty in text classification (more detail in Subsections II-E and III-E3). Reference [16] bring a similar argument against b) that adversarial examples are often overly synthetic and disconnected from real-world performance concerns, which we assert to be especially true for perturbations applied to text data. Therefore, we derive a challenging experimental setup for b) (more detail in Subsection III-E2) inspired by the extensive literature in NLP on the problem of domain shifts and domain adaptation [17]–[22]. Domain adaptation approaches aim to mitigate performance degradation that occurs when transferring a classifier from a source domain to a target domain. Learning under domain shift presents a complex challenge in text classification since linguistic patterns can be highly different across domains, even harder to tackle when domains are unknown a priori [22]. While out-of-domain generalization is the ultimate objective [23], we believe that accurate uncertainty prediction has a major role to play in the detection of out-of-domain data, which is currently under-explored. Reference [24] is a notable exception where predictive uncertainty methods are leveraged to learn domain-invariant features in unsupervised fashion.

In this work we only consider methods that directly estimate the predictive posterior and aim at obtaining high quality uncertainty estimates by discriminative models without any additional OOD components. However, there exists a large number of alternative OOD detection and generalization approaches. We surmise that these can be more effective in handling the above distribution shifts, yet they have different modeling assumptions which complicates a direct comparison, for instance, access to (auxiliary) OOD data [25], [26], generative modeling [27], focus on abstention mechanisms [28], or characterization of dataset shifts with a two-sample-testing approach [29]. We recommend [30], [31] for an overview of these approaches.

While previous BDL benchmarks have helped standardize protocols, metrics and analysis tools, the effort is not spent equally across all modality and problem settings (as can be observed in the survey of [32]). Arguably, most research on uncertainty estimation focuses on regression and image classification tasks as they offer visual validation on uncertainty quality, e.g., [33].

Tasks in the NLP field involve discrete natural language units (word, sentence, paragraph) as input, which requires a translation to the continuous domain by embedding discrete units to form high-dimensional distributed representations [34]. This presents additional complexity compared to image or time-series data which as continuous signals can be directly fed into a Neural Network (NN). Furthermore, specialized algorithms (e.g., dealing with long sequences, attention for larger memory [35]) and progressively more

complex architectures [36] are being created to tackle this unique challenge in NLP, which can affect the performance of predictive uncertainty techniques. With our work, we start the exploration into effects of field characteristics, notably different NLP architectures, inherent task complexity, and properties of language in text processing (e.g., ambiguity [37], document length [38], pre-defined vocabulary [39]) that could cause problems when predicting uncertainty. More specifically, we seek to answer how uncertainty research translates to a prototypical language task such as *text classification*, which more frequently than vision tasks is characterized by non-mutually exclusive labels [40], a problem setting ignored by existing BDL benchmarks.

BDL research on NLP tasks is generally limited, certainly when considering quantitative evaluation of predictive uncertainty quality. While we draw inspiration from the uncertainty estimation methods of [41], their study focuses on the performance increase of non-probabilistic measures (mean-squared error) and only reports sentiment regression results. Moreover, we find no quantitative evaluation of the quality of the uncertainty scores and comparison to simpler measures of uncertainty, for instance, softmax score or predictive entropy. [42] does focus on the robustness of pre-trained Transformers to distribution shift, yet without application of any predictive uncertainty methods. References [43], [44] present similar setups applying Monte Carlo Dropout to regular NLP architectures in an active learning setup, yet they only aim to increase overall predictive performance by relying on in-domain calibration. Our work benchmarks individual and joint predictive uncertainty methods in multiple text classification task settings over two well-motivated uncertainty evaluation setups, testing robustness to distribution shift for NLP problems.

II. UNCERTAINTY METHODS

The first Subsection formally presents how to quantify uncertainty in BDL and how popular methods approach inference differently. Subsection II-B treats predictive uncertainty methods with a focus on the algorithmic procedure, followed by representative method extensions for more reliable uncertainty estimation. Subsection II-C describes from what sources uncertainty originates and how to quantify uncertainty at test-time. In Subsection II-D we present the rationale of our study, connecting recent research on how NNs navigate the optimization landscape with the posterior approximation procedure of methods from Subsection II-B. Subsection II-E provides a critical note on how distribution shift impacts uncertainty estimation and the evaluation thereof.

A. QUANTIFYING UNCERTAINTY IN DEEP LEARNING

In modern Deep Learning, two common uncertainty (or inversely “confidence”) estimates are the maximum posterior class probability, known as *softmax-score*, and the *predictive entropy* over posterior class probabilities [45], [46]. However, [47]’s work on confidence calibration demonstrated these to be unreliable estimates of Neural

Networks' uncertainty. While post-hoc calibration methods such as Temperature or Vector Scaling [47], [48] can easily calibrate classifier uncertainty in-domain (further discussed Subsection II-E), they have been found to be less effective under increasing distribution shift [12], [14].

Bayesian Deep Learning (BDL) methods build on solid mathematical foundations and hold promise for more reliable learned uncertainty estimates [7]. Drawing on the ground-laying works of [49]–[53], the “second-generation” in BDL [54] is geared towards finding practical and scalable approximations to the analytically intractable Bayesian posterior (Eq. 1). Inferring a prediction and the associated uncertainty for a new test input x^* (with its associated label vector y^*) requires computing the conditional probability of x^* given the training data $\mathcal{D} = \{(x^{(n)}, y^{(n)})\}_{n=1}^N$,

$$P(y^* | x^*, \mathcal{D}) = \int P(y^* | x^*, \mathcal{D}, \theta) \underbrace{P(\theta | \mathcal{D})}_{\text{posterior}} d\theta, \quad (1)$$

with θ representing all Bayesian Neural Network (BNN) parameters: weights w , biases b .

In our study we will focus on two strategies with representative methods that circumvent the *inference problem* and have seen more widespread adoption given their ability to scale both in network architecture and dataset size.

I. The weight snapshots direction, Deep Ensemble [55], which aims to find different sets of model parameters. Snapshots can be collected during different stages of training [13], [56], [57], or by using a sampling process such as Markov Chain Monte-Carlo (MCMC) [58]–[60]. **II. The stochastic computation-graph direction, Monte Carlo Dropout** [61], involves introducing noise over weights during training and estimating uncertainty with multiple stochastic forward passes. Recent works [62], [63] have proposed “single-model” uncertainty methods that ideally compute posterior uncertainty in one forward pass.

Our work benchmarks representative methods from both categories (denoted by cursive), motivating a cross-category comparison and analyzing their individual-joint effectiveness in modeling predictive uncertainty.

Additionally, we experimented with alternative scalable uncertainty methods, namely stochastic gradient MCMC methods, *cyclical SG-MCMC* (cSG-MCMC) [60], and a single forward pass uncertainty method incorporating a Gaussian Process (GP) output layer, *Spectral-normalized Neural Gaussian Process* (SNGP) [63]. Results and discussion for these are included as self-contained Subsections in Appendix C.

B. PREDICTIVE UNCERTAINTY METHODS

We will first introduce each method by explaining the algorithm, followed by advantages or identified shortcomings, with subsequent method extensions from the same procedure category. Finally, we will zoom in on how to quantify uncertainty using each method.

1) MONTE CARLO DROPOUT

The seminal work of [61] on Monte Carlo Dropout (MC Dropout, MCD) proposes efficient model uncertainty estimation by exploiting dropout regularization as an approximate Variational Inference (VI) method. In practice, the MCD procedure boils down to (i) applying dropout on all non-linear layers' weights, and (ii) activating dropout both during training and evaluation. Quantifying “epistemic” *model uncertainty* using MCD involves sampling T stochastic weight sets from the variational Bernoulli distribution $\hat{\theta}_t \sim q(\theta)$ to calculate the lower-order moments of the approximate Gaussian posterior, respectively the predictive mean and variance (Eq. 2).

$$\begin{aligned} \hat{\mu}_{pred}(x^*) &= \frac{1}{T} \sum_{t=1}^T P(y^* | x^*, \hat{\theta}_t), \\ \hat{\sigma}_{pred}^2(x^*) &= \frac{1}{T} \sum_{t=1}^T [P(y^* | x^*, \hat{\theta}_t) - \hat{\mu}_{pred}]^2 \end{aligned} \quad (2)$$

MCD's simplicity and computational tractability, i.e., dropout training is a standard DL practice and prediction only requires 1 model to sub-sample from, has made it one of the most popular predictive uncertainty methods. However, an important shortcoming of VI, and in consequence MCD in [61]'s formulation, is that it is known to underestimate predictive variance [64]. We will touch on a selection of method extensions in further subsections (II-B3, II-B4).

2) DEEP ENSEMBLE

Deep Ensemble [55] (DE) involves independently training multiple probabilistic NNs with different random weight initializations and aggregating predictions from individual models. An ensemble of NNs trades off computational resources, due to the need to train and store M models, for uncertainty estimation and robustness to dataset shift [12], [65], [66]. In comparison to MC Dropout, DEs are treated as a uniformly-weighted Gaussian Mixture model, to which the formula for predictive variance is adapted:

$$\begin{aligned} \hat{\sigma}_{pred}^2(x^*) &= \frac{1}{M} \sum_m \left(\sigma_{\theta_m}^2(x^*) + \mu_{\theta_m}^2(x^*) \right) - \mu_*^2(x^*), \\ \mu_*(x^*) &= \frac{1}{M} \sum_m \mu_{\theta_m}(x^*) \end{aligned} \quad (3)$$

The empirical performance increase of ensembles can be attributed to the diversity of uncorrelated errors between ensemble members [67]. Without functional diversity in sets of model parameters, posterior approximation quality will be lower (zero variance) and for this reason, ensemble diversity promotion is a promising avenue for further improvements [68], [69]. Alternatively, the interplay between ensembling and regularization, “the effect of a prior”, warrants more thought, since not regularizing risks overfitting, while too strong regularization risks constraining diversity (see Subsection II-D).

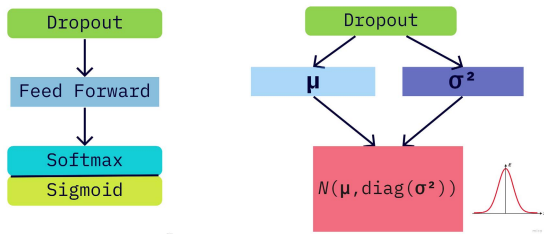


FIGURE 1. Visualization of output layer blocks. The left block denotes standard softmax (multi-class) or sigmoid (binary/multi-label) output. On the right, the heteroscedastic model outputs a normal distribution $\mathcal{N}(\mu(x), \text{diag}(\sigma^2(x)))$ parametrizing mean and variance by the logits coming from two separate preceding feedforward layers.

3) CONCRETE DROPOUT

Reference [70] proposes a **Continuous-discrete** distribution relaxation to adapt and optimize the dropout probability p as a variational parameter using standard gradient descent. This overcomes the limitations of uncertainty underestimation, miscalibration, and the computational complexity of manually tuning layer-wise dropout probability in deeper models [71]. By taking advantage of the reparametrization trick, the Concrete distribution approximation \tilde{z} of the original Bernoulli random variable z conveniently parametrizes to a simple sigmoid distribution ($\phi = \text{sigmoid}$) allowing for gradient-based optimization. Given a uniform random noise variable u and a temperature r , the expression varies with respect to the dropout probability p , which for $p \rightarrow 0.5$ produces by a rate of $\frac{1}{r}$ values approaching 1.

$$\tilde{z} = \phi\left(\frac{1}{r}(\log p - \log(1 - p) + \log u - \log(1 - u))\right) \quad (4)$$

Since the dropout probability characterizes the overall posterior uncertainty, Concrete Dropout can positively influence in-domain calibration at an almost negligible cost.

4) HETEROSCEDASTIC EXTENSIONS

References [41], [72], [73] proposed similar approaches to extend MC Dropout to allow measuring uncertainty information from different sources. Estimating input-dependent, “heteroscedastic aleatoric”, *data uncertainty* (detail Subsection II-C3) requires slightly modifying the model’s architecture and objective function following [72].

Firstly, the output layer of model $f_{\hat{\theta}}$ is extended with a set of learnable variance variables σ^2 per unique class output. The model’s output logits, \mathbf{v} , are sampled from the stochastic output layer parametrized by $\mathcal{N}(f_{\hat{\theta}}(x), \text{diag}(\sigma^2(x)))$. This model adaptation will be referred to as the *heteroscedastic model*. Fig. 1 visualizes the difference in output layer design.

Next, it requires incorporating a *heteroscedastic loss*:

$$\mathcal{L}_{\text{HET}}(\hat{\theta}) = \sum_{i=1}^N \log \frac{1}{T} \sum_{t=1}^T \exp\left(\mathbf{v}_{i,c}^{(t)} - \log \sum_k \exp \mathbf{v}_{i,k}^{(t)}\right) + \log T \quad (5)$$

with N the number of training examples passing through an instance t of the model $f_{\hat{\theta}_t}(x) + \sigma^{(t)}$ ($^{(t)}$ omitted for sampling superscript) to generate for example i a sampled logit vector $\mathbf{v}_i^{(t)} \in \mathbb{R}^K$, where predicted value for class k , $\mathbf{v}_{i,k}^{(t)} \in \mathbb{R}$, and c the index of the ground truth class. The above loss formulation shares notation with a categorical cross-entropy objective, although the loss is computed over T sampled logits $\mathbf{v}_i^{(t)}$ perturbed with parameterized Gaussian noise. By learning to predict log variance over T dropout-masked samples, the model will be able to output high variance (uncertainty) for inputs where the predictive mean is far removed from the true observation, which by design has a smaller effect on the total loss.

C. UNCERTAINTY ESTIMATION

In this Subsection, we will introduce sources of uncertainty, a categorization of uncertainty measures, and how uncertainty is quantified in practice.

1) TOTAL UNCERTAINTY

Classification models trained by minimizing negative log-likelihood quantify global uncertainty over class outcomes with entropy (H) over logits. Therefore, the entropy of the posterior predictive distribution provides a measure of the total uncertainty, which is a combination of model and data uncertainty [74]. Instead of entropy, posterior predictive variance can also be decomposed into model and data uncertainty using the law of total variance [75]. Decomposing total uncertainty into the different sources is beneficial for determining actions to evaluate the room for improvement.

2) MODEL UNCERTAINTY

Epistemic uncertainty presents the inherent ignorance [71] of the model with regards to the true values for its parameters and structure after having seen the training data. Next to predictive variance, *Mutual Information* (MI) [76] has been proposed as a measure of epistemic uncertainty, as intuitively it captures the amount of information that would be gained about model parameters through “knowledge” of the true outcome [77].

3) DATA UNCERTAINTY

Aleatoric uncertainty captures the inherent stochasticity and noise in data. It can be further decomposed into a *homoscedastic* component, which represents constant noise over inputs such as the numerical accurateness of a measuring device, and *heteroscedastic* uncertainty representing input-dependent noise generated by class overlap, complex decision boundaries or label noise [75]. Heteroscedastic data uncertainty allows for the expression of instance-level uncertainty together with the best possible prediction.

4) UNCERTAINTY CATEGORIZATION

Here follows a categorization of the uncertainty measures from methods (and combinations) of Subsection II-B.

We directly provide estimators for the theoretical quantities that are defined as either arising from entropy or variance-based uncertainty decomposition in [75]. To estimate for a new test sample x^* the prediction and uncertainty of model $f_{\hat{\theta}}(x^*)$ we typically seek to obtain the predictive posterior distribution $P(y^*|x^*, \hat{\theta})$ over class membership probabilities with $y_k^* \in \{1, \dots, K\}$.

For MC Dropout at inference time, we presume $P(y^*|x^*, \hat{\theta}) \approx \frac{1}{T} \sum_{t=1}^T P(y^*|x^*, \hat{\theta}_t)$, with prediction obtained after applying softmax/sigmoid function for sample t , $\hat{p}_t = P(y^*|x^*, \hat{\theta}_t)$. For Deep Ensemble, the above notations would require a change from T to M , but for consistency over quantity formulas, we maintain T to denote posterior sampling. For ease of notation, we define a helper entropy function on $H(x^*, \cdot) = -\sum_{k=1}^K P(y_k|x^*, \cdot) \log P(y_k|x^*, \cdot)$ with \cdot an input argument to the function.

Quantity	Formula
Softmax-score	$S = \max_k \frac{\exp f_{\hat{\theta},k}(x^*)}{\sum_{j=1}^K \exp f_{\hat{\theta},j}(x^*)}$
Predictive Entropy	$H_{pred} = H(x^*, \hat{\theta})$
Mutual Information	$I = H_{pred} - \frac{1}{T} \sum_{t=1}^T H(x^*, \hat{\theta}_t)$
Model Uncertainty	$\hat{\sigma}_{model}^2 = \frac{1}{T} \sum_{t=1}^T (\hat{p}_t - \hat{\mu}_{pred})^2$
Data Uncertainty	$\hat{\sigma}_{data}^2 = \frac{1}{T} \sum_{t=1}^T \frac{1}{K} \sum_{k=1}^K \text{var}_k^{(t)}(x^*)$

For any classification model, it is possible to compute the softmax-score and predictive entropy. For multi-label classification, the softmax-score does not take into account multiple winning classes and a standard approximation² would be to average over the sigmoid-scaled probabilities of predicted classes.

Model uncertainty can be quantified with Monte Carlo integration or the aggregation of individual models [78]. In practice, it is quantified by either (a) calculating the average sigmoid/softmax variance over the predictive mean from MC samples (Eq. 2) or (b) computing the total variance from an ensemble mixture distribution (Eq. 3). Changing to the heteroscedastic extensions allows to quantify data uncertainty. More specifically, data uncertainty is quantified with as “surrogate” [41] the average over variance logits $\text{var} = \sigma^2$ (see Fig. 1). Whenever ensembling is applied where a single model estimates a quantity, one typically averages over the ensemble components’ uncertainty.

²Intending to compare directly with multi-class results, averaging uncertainty estimates to obtain a single summary statistic for multi-label predictions is more straightforward than reporting class-wise results. In particular, the tested multi-label datasets share low average label cardinality, a high degree of label correlation, and a large set of unique classes ($K > 50$).

D. MOTIVATING HYBRID APPROACHES

This Subsection will motivate the theorized complementarity of VI-based and ensembling methods for improved uncertainty estimation and robustness.

In light of the empirical success of Deep Ensemble, recent research [7], [79] raises an important question concerning the difference in function-space between variational Bayesian NNs (MC Dropout and extensions) and Deep Ensemble. Deep NNs are parametrized (typically non-linear) functions presenting a high-dimensional non-convex optimization problem, which may concern widely varying curvature and many flat regions with multiple locally optimal points within each [80]. Applying an optimization procedure to a maximum-a-posteriori (MAP) objective involves a search for parameter values (*hypotheses*) for which the loss function is low by navigating the high-dimensional loss landscape. Once model training converges, one ends up with a weight-space *solution*, representing a single *mode* of the parameter posterior. One such mode is a local optimum of the loss function $\mathcal{L}(\theta)$, representing unique functions $f_{\hat{\theta}}$ as a set of NN parameters [57]. Each mode potentially marks a meaningfully different representation of the data.

The true posterior is generally a highly complex and multimodal distribution, with multiple possible but not necessarily equivalent parametrizations θ able to fit the training data. To accurately quantify posterior uncertainty, we wish to capture as many modes or separated regions as possible [7], [81].

Correspondingly, the common goal is to achieve reliable uncertainty and, following the BDL paradigm, one resorts to modeling a Bayesian posterior. What differs among the selected predictive uncertainty methods, is the form of the prior $P(\theta)$ over model parameters and likelihood $P(\mathcal{D}|\theta)$ [82], from which to determine a procedure. Below we expound on the **difference in posterior approximation procedure**:

- MC Dropout is a common VI procedure with Bernoulli dropout and Gaussian (L2) priors on weight-space, assuming a posterior Gaussian distribution from which to draw stochastic samples. VI-based methods tend to locally approximate uncertainty surrounding a single mode, **intra-modal** posterior approximation. Specifically, MC Dropout’s procedure can be interpreted as imposing a spike-and-slab parameter prior with peaked variance [83], which offers a plausible explanation for approximated uncertainty centered tightly around 1 mode.
- An ensemble of NNs makes no direct assumptions on the form or distribution of the prior and just “obtains” different samples from the parameter posterior. It generates a series of MAP estimates which through inherent stochasticity in weight initialization and optimization end up at different regions in weight space, leading to functionally dissimilar but more or less equally accurate modes of the solution space. Due to randomness in the optimization, some solutions may be significantly worse than others as measured by different metrics

(e.g., accuracy vs. calibration). Ensembles are effective at exploring the weight-space and by solving the MAP estimation problems converge to multiple modes [81], [84], allowing for **inter-modal** posterior approximation. Furthermore, by considering more possible hypotheses they will be better at approximating multi-modal posterior distributions and avoid the collapse to a single mode [7].

Combining both procedures is to generate a mixture over priors [85], which in itself is again a prior, all under the same likelihood function. There is no guarantee that a combination of methods from both procedures captures the true posterior, yet in our work we will empirically analyse if combining inter and intra-modal posterior approximation offers the hypothesized complementary benefits.

E. UNCERTAINTY CALIBRATION UNDER DISTRIBUTION SHIFT

In this Subsection, we motivate the meaningfulness of evaluating uncertainty methods under distribution shift and what restricted assumptions one should reasonably specify to guarantee useful empirical results.

We consider the problem of detecting out-of-distribution data from a trained classifier's uncertainty. Let $P^S(x, y)$ and $P^T(x, y)$ denote two distinct distributions, respectively *in-domain* and *out-of-domain*. Further we assume the classifier $f \rightarrow [0, 1]$ trained on P^S , whereas in the experimental setup we test on a mixture distribution $\mathbb{P}^{(S,T)}(x, y)$. Given an input x from the mixture, we test if the classifier's uncertainty can be exploited to distinguish from which distribution the sample comes. To be clear, in this setting we expect to detect uncertainty arising from distribution shift and not from a lack of training data. It can be argued that there is a relationship between both, as having few in-domain samples complicates generalization, in turn increasing the chance of flagging a new data point as OOD.

Uncertainty estimation is generally well-defined in the context of in-domain data with the standard assumption that samples are independent and identically distributed (IID). In this setting, evaluation is typically expressed in terms of **calibration** (Def. 1), particularly as statistical error with respect to the conditional expectation (Def. 2).

Definition 1 (Perfect Calibration [86], [87]): Calibration is a property of an empirical estimator f , which states that on finite-sample data it converges to a solution where the scoring function reflects the probability v of being correct. Perfect (in-domain) calibration, $\text{CE}(f, P^S) = 0$, is satisfied if:

$$\mathbb{P}(\hat{Y} = Y \mid f(X) = v) = v, \quad \forall v \in [0, 1]$$

Definition 2 (Calibration Error [88], [89]): The ℓ^p calibration error of $f : \mathcal{X} \rightarrow [0, 1]$ under distribution Z over $(X \times Y)$ with the norm $p \in [1, \infty)$ is given by:

$$\text{CE}^p(f, Z) = \mathbb{E}_{(x,y) \sim Z} \|\mathbb{P}[\hat{y} = y \mid f(x)] - f(x)\|^p \quad (6)$$

To obtain a reliable probabilistic classifier in the traditional IID setting, explicit in-domain re-calibration approaches are

effective [47], [90], [91]. However, there is no general principle which states that a classifier, however calibrated on P^S , would be calibrated on OOD data from P^T . Infinitely many possible shifts can violate the standard IID assumption at varying degrees of severity, affecting calibration and uncertainty estimation in unpredictable ways. With the aim of still being able to rely on a classifier's uncertainty calibration to predict future generalization, there is a need to relax the IID assumption. An important condition for meaningful uncertainty estimation is to impose realistic, yet sufficiently restrictive assumptions on the nature of distribution changes and how P^S and P^T relate. The **covariate shift** [92], [93] assumption may be the most widely studied when the real-world data distribution differs from the training distribution.

Recently, [94] formalized the problem of calibrated prediction under covariate shift with theoretical bounds on calibration transfer over domains. Critically, related works [95]–[99] prove with importance weighting that shared structure and high overlap in distribution support (or conversely, low domain divergence) is crucial to upper bound the increase of calibration error due to covariate shift. To put it plainly: while one cannot guarantee calibration on OOD data in the general case, if domains are reasonably close one can expect to retain (some if not most) benefits from in-domain calibration.

Specific to our work, we consider two experimental settings (Subsection III-E) with different distribution shift [100] between domains. Here we characterize each with the related distribution shift assumptions. (i) *Cross-domain classification*, where covariates differ $P^T(X) \neq P^S(X)$, but label distributions are identical $P^T(Y|X) = P^S(Y|X)$ [92]. (ii) *Novelty detection*, where label distributions disagree $P^T(Y|X) \neq P^S(Y|X)$, since the label sets differ between domains $[Y]^T \neq [Y]^S$ [101]. Whereas (i) is a clear case of covariate shift, we reasonably assume for (ii) that covariates are generally close $P^T(X) \approx P^S(X)$ and that the overall conditional shift will be small. Rather than interpreting novelty as a shift in label sets, one might define the probability of seeing some labels under S as exactly zero, while under T their probability is $\varepsilon > 0$. In practical text classification settings, novel class inputs will typically start occurring with small frequency in the real-world data distribution, as well as not having completely different syntax and semantics. This implies that 'excess' calibration error (defined as an expectation over the mixture) will only be impacted slightly.

Clearly specifying distribution shift assumptions is quintessential for reliably benchmarking uncertainty methods, since the calibration of each tested method can be affected in different ways and produce results biased towards an evaluation configuration. In our selected experimental settings, we can justify uncertainty calibration under distribution shift as a reasonable methodology, without making further claims on the general applicability of this evaluation procedure.

III. EXPERIMENTAL METHODOLOGY

In this work, our objective is to reliably benchmark both existing and novel combinations of predictive uncertainty methods in order to draw conclusions for text classification applications. This Section describes our study's experimental methodology with which we generate the empirical evidence presented in Section IV. Subsection III-A introduces our hypotheses on complementary benefits for uncertainty estimation and details the hybrid methods. Provided the focus on text classification tasks, Subsection III-B motivates a set of representative datasets, with a specification of different text problem characteristics. Subsection III-C documents two pre-selected text classification architectures, the first a simple and more controllable configuration for uncertainty benchmarking, the second a more complex NLP architecture for which we will compare relative gains in robustness. To ensure correct performance benchmarking, Subsection III-D summarizes the metrics used for evaluating calibration and robustness. Finally, Subsection III-E expounds on the model setups and experimental settings devised to compare predictive uncertainty methods.

A. PROPOSED HYBRID APPROACHES

This Subsection stands central in our work in which we motivate combinations of predictive uncertainty methods. We build hypotheses on complementary benefits from combining multiple uncertainty methods, for which we present an overview of hybrid methods in scope of our experiments (Table 1).

Given the obvious parallels and differences between both procedures presented in Subsection II-D, we hypothesize **complementary benefits** for uncertainty estimation and robustness.

- A. Whereas ensembles are adept at capturing multiple modes, they do not approximate uncertainty surrounding a single mode in solution space. However, since there is a lot of redundancy in function space, local neighborhood uncertainty approximation might make only a minimal contribution to the overall posterior uncertainty. [79] validated that applying subspace sampling on an optimized solution improves in-domain accuracy and calibration. They note improvements relatively lower than increasing ensemble size (M), yet they did not analyze for joint effectiveness.
- B. A procedure can only be as good as the prior and the likelihood function, which in approximation of the intractable parameter posterior is limited by computational constraints (number of MC samples T , number of ensemble models M). By lack of any specific prior constraining the optimization of independent ensemble members, the regularization effect from VI-based priors such as dropout may introduce smoothness [102], [103], inducing a simpler optimization landscape with less (possibly weak) hypotheses present. In turn, by modeling an ensemble of VI approximate posteriors less ensemble members could be required to reach the

TABLE 1. In total, we consider 18 model setups, based on combining methods and options from each column. (*) Deterministic dropout can only combine with Deep Ensembles. CE stands for cross-entropy loss.

Dropout	MC sampling	Heteroscedastic	Deep Ensemble
$p = 0^*$	$T = 1$	\mathcal{L}_{CE}	$M = 1$
$p = 0.5$	$T = 10$	\mathcal{L}_{HET}	$M = 5$
Concrete			

same in/out-of-domain performance as measured by the size and quality of captured solutions. Reference [79] already observed that ensembles saturate after reaching peak in-domain performance, with suboptimal models taking over the benefit.

- C. Important to note is that the influence of the prior and variational parameters requires fine-tuning, since over-regularization will reduce the optimization problem to one with an over-smooth, possibly unimodal landscape [57], [81]. This eliminates any functional diversity for whatever ensemble size, where the solution will be overconfident. Alternatively, since the hypothesis space for a NN is often so large, with many possible likely models for finite data, that some posterior collapse will often be desirable to reduce the number of considered hypotheses. [7].

Table 1 summarizes all model setups and hybrid methods considered for our experiments. The most complete combination is *MC Concrete Dropout Heteroscedastic Deep Ensemble*, where each member m of the ensemble has optimized the layer-wise dropout rate p and heteroscedastic loss \mathcal{L}_{HET} , with the final predictive distribution over K classes deriving from M times T stochastic MC Dropout samples ($M \times T \times K$).

We admit two baselines, *Unregularized* and *Regularized*.

Unregularized ($p = 0$) offers a clean comparison, discounting any influence of sparsification (dropout) or normalization of weight magnitude (weight decay). However, it possibly overfits parameters to training data. In practice, one would always apply some combination of regularization (dropout, weight decay, batch normalization, data augmentation, ...) to counter overfitting. *Regularized* ($p = 0.5$) gives an alternate point of comparison over uncertainty methods, such that we can exclude that performance increase for an uncertainty method does not only come from regularization, which some such as MC Dropout rely upon.

Adhering to good practices and since we build ensembles with default $M = 5$, we report the mean (and standard deviation) for all individual models, making the results more statistically reliable than comparing to 1 independently trained model.

B. DATASETS

We use six well-studied real-world text corpora characterized by a different number of classes, classification task, and size of the documents (Table 2).

TABLE 2. D denotes the number of documents in the dataset, K the number of classes, I the class imbalance ratio [104], W the average number of words per document, V the total vocabulary size respectively.

corpus	task	D	K	I	W	V
20news	newswire topic	18,848	20	5e-4	240	212,267
IMDB	movie review	348,415	10	0.03	325.6	115,073
CLINC-OOS	intent detection	22,500	150	0	8	6,188
Reuters ApteMod	newswire topic	10,786	90	0.14	125.2	65,035
AAPD	academic paper subject	55,840	54	0.04	145.4	66,854
Amazon Reviews (#4)	product sentiment	8,000	2	0	189.3	21,514

The first three datasets share the task of multi-class classification in three different text domains.

20News [105] is a collection of 20K newsgroup documents with balanced samples for 20 different newsgroups. To allow for direct comparison, we use the dataset in the benchmark format of [106].

IMDB movie reviews [107] (imdb) is a large sentiment classification dataset which links user-based reviews of movies with labels on an ordinal scale between 1 and 10. Since there are no standard splits for this dataset we generate randomized (seed 42) stratified splits of 65% for training, 15% validation and 20% for testing.

CLINC-OOS (CLINC150) [108] is a recently become popular intent detection dataset comprising 150 training sentences for each of the 150 system-supported services. Next to this, it offers a separate Out-of-Scope (OOS) subset with 1200 natural sentences which can be used for Out-of-Domain (OOD) detection, more specifically detecting novel class instances. This dataset differs from the previous two through very short “intent” sentences requiring classification in a large output space. For training and evaluation, we use the predefined splits of TensorFlow Datasets.

We include two popular multi-label text classification datasets, since they are often not considered for uncertainty experiments. We argue that they should be included since their multi-label nature is very common in text classification where not all labels have to be mutually-exclusive, e.g., topic categorization, subject attribution, ...

Reuters ApteMod [109] is a multi-label news topic categorization dataset with 90 possible topics and an average low label cardinality (C) of 1.24. We use the standard ApteMod splits.

Arxiv Academic Paper Dataset (AAPD) [110] comprises 55,840 computer science paper abstracts that have been labeled with corresponding multiple subject matters. Each academic paper has on average 2.41 subject targets with a minimum of 2. For reproducibility purposes, we use the same preprocessing steps and splits as in [110], [111] with 1K dev and 1K test samples.

Amazon Reviews [17] is a widely-used benchmark for domain adaptation research in NLP. It consists of binary sentiment classification datasets from four different domains: Books, DVDs, Electronics and Kitchen appliances. Each domain dataset contains 1K positive and 1K negative labeled instances. Following the convention of previous works [21], [112], we construct 12 balanced cross-domain sentiment

analysis tasks, where for each source dataset we randomly hold out 400 test instances to evaluate in-domain and always predict on the full target dataset. We reserve this dataset for cross-domain experimentation only (Subsection III-E2).

C. ARCHITECTURE

This Subsection motivates the two NLP architectures in scope for the experiments.

1) TEXTCNN ARCHITECTURE

We use a 1-D Convolutional NN for text classification (TextCNN), following the model structure of [113]. We chose this architecture for its comparative simplicity and solid out-of-the-box performance on a range of text classification tasks. Even as a light-weight model, it can deal with feeding in text sequences of varying sizes and learning n-gram-like structures over word embeddings, allowing a fair comparison across text datasets. An extensive hyperparameter study determined that regularization does not impact performance much [114].

2) TRANSFORMER ARCHITECTURE

Models in NLP have become increasingly deeper and more complex with the advent of the Transformer architecture [35]. [115] have combined multiple bidirectional Transformers with wordpiece tokenization and self-supervised pre-training objectives —masked language modeling and next sentence prediction— to create the contextual representation modeling architecture BERT. It allows for fine-tuning on downstream tasks where BERT has outperformed task-specific architectures even in low resource settings. In our experiments we use BERT_{base} (uncased, English): 12 layers, 768 hidden dimensions, 12 attention heads, with a total number of 110M parameters.

3) COMPLEXITY

TextCNN comprises only 6M parameters with most parameters residing in the embedding matrix. However, it is restricted to a fixed window size with the downside of not being able to determine long-distance dependencies in text. BERT, on the other hand, has already captured prior language modeling knowledge thanks to pre-training. Nevertheless, our experiments already involve significant computational complexity, which is why we decided not to run all variations with BERT. TextCNN presents a more controllable configuration, achieving decent performance and satisfying for the evaluation of predictive uncertainty in text classification. We include an ablation study (Subsection IV-D2) comparing specifically selected models trained with BERT as base architecture.

D. EVALUATION METRICS

Since no single metric measures all desirable properties of predictive uncertainty, we use a variety of conventional metrics to evaluate our models’ performance, (a) *calibration metrics*, b) *proper scoring rules* and c) *classification scores*.

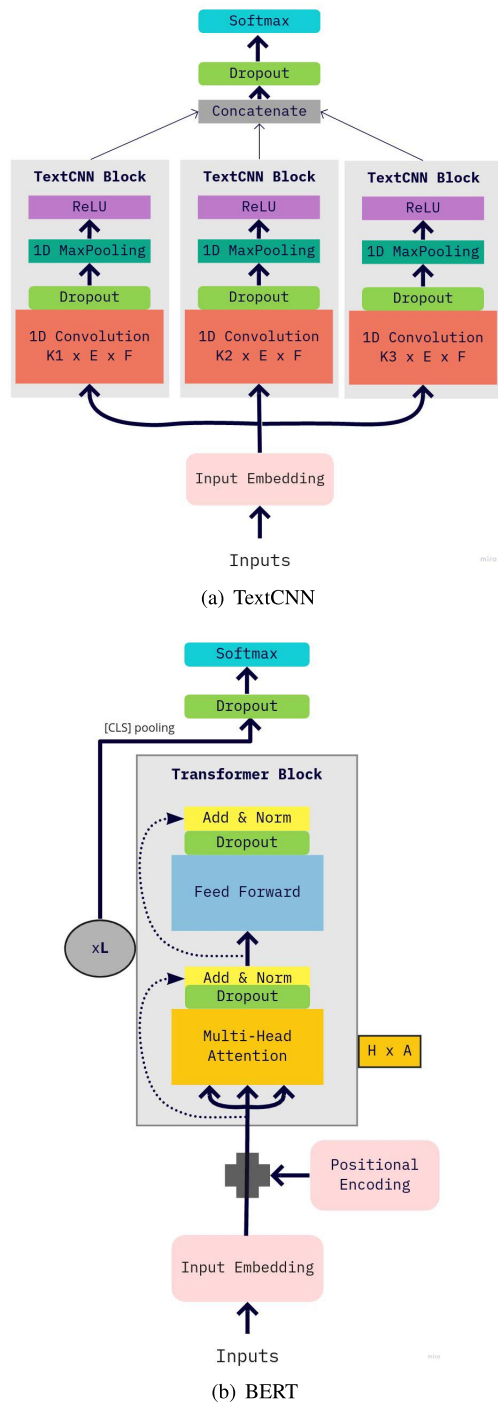


FIGURE 2. Simplified block-diagrams for each of the NN architectures, demonstrating on which layer weights dropout is applied. (a) The TextCNN model architecture with 3 kernels ($K_1 - 3$), E word embedding dimensionality and F number of feature maps per kernel. (b) The BERT model architecture with L Transformers blocks, hidden size H and number of self-attention heads A .

For **in-domain evaluation**, we use the following metrics:

- (a) **Expected Calibration Error (ECE)** [47], [88] is an intuitive metric often used in practice to score the calibration of maximum posterior predicted probabilities. This metric separates the probability space in B bins

where for each bin the gap between observed accuracy and bin confidence \bar{P}_b is measured, with a final average weighted by number of samples per bin $|b_i|$.

$$ECE = \sum_{i=1}^B \frac{|b_i|}{N} |acc(b_i) - \bar{P}_b(b_i)| \quad (7)$$

Alternative (theoretical) formulations have been developed for multi-class [116], [117] and multi-label calibration [87], [118]. Measurements of “strong” calibration, over the full predicted vector instead of the winning class, are reported less in practice. Possible reasons are that they render class-wise scorings, potentially based on adaptive thresholds, or require estimation of kernel-based calibration error to derive hypothesis tests. While we are mindful of alternatives, “weak” calibration measured by ECE meets the practical requirements for our benchmarking.

Next to calibration, we turn to *proper scoring rules* [119], which calculate scoring at the instance-level while measuring both the quality of accuracy and calibration (decomposable into refinement and calibration losses [120]).

- (b) **Negative Log Likelihood (NLL)** [121] is both a popular loss function (*cross-entropy*) and scoring rule which only penalizes (wrong) log probabilities q_i given to the true class, with \mathbb{I} an indicator function defining the true class. This measure more heavily penalizes sharp probabilities, which are close to the wrong edge or class by over/under-confidence.

$$\mathcal{L}_{NLL}(\mathbf{Q}, \mathbf{y}) = -\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K \mathbb{I}(y_i = k) \cdot \log(q_{i,k}) \quad (8)$$

- (b) **Brier Score** [122] is a scoring rule that measures the accuracy of a probabilistic classifier and is related to the mean-squared error loss function (*MSE*). Brier score is more commonly used in industrial practice, since it is an ℓ^2 metric (score between 0 and 1), yet it penalizes tail probabilities less severely than NLL.

$$\mathcal{L}_{BS}(\mathbf{Q}, \mathbf{y}) = \frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K (\mathbb{I}(y_i = k) - q_{i,k})^2 \quad (9)$$

For **distribution shift evaluation**, we use binary classification metrics following [106], (c) **AUROC** and **AUPR**, both threshold-independent measures with the latter accounting better for class imbalance, which we use to summarize detection statistics of positive (out-of-domain) versus negative (in-domain) instances. When evaluating a model trained in a source domain on a target domain with a similar task, we denote accuracy in the target domain as **OOD accuracy** as opposed to accuracy in the source domain, which we denote as **ID accuracy**.

E. EXPERIMENTAL DESIGN

We have determined three logical settings in text classification to evaluate predictive uncertainty for each model setup. We present experiments on in-domain uncertainty to form baseline results, followed by cross-domain classification with

a focus on out-of-domain detection, and finally we propose novelty detection as a new protocol to evaluate predictive uncertainty.

While there is no gold standard procedure for comparing multiple (uncertainty) methods over multiple (text classification) datasets, we opted for an established procedure with statistical testing via multiple comparisons [123], [124]. Since we present an exhaustive list of model setups, we present our results in terms of rank and critical difference diagrams in order to analyze relative performance of each method over different experimental settings.

Concretely, each dataset concerns independent measurements, for which we rank each method, then compare average ranks, and in the event that we can reject the null-hypothesis (\mathcal{H}_0 : all methods have the same rank), we calculate post-hoc tests with critical differences over methods. However, only reporting ranks does not allow future researchers to compare to our work, which is why we include detailed absolute number results in the Appendix D.

1) IN-DOMAIN SETTING

To evaluate in-domain (ID) uncertainty, we will focus on measuring calibration and prediction quality with proper scoring rules (see Subsection III-D). The ID setting assumes that the train and test examples are independent and identically distributed (IID). To capture all details, we compare per task-setting, multi-class and multi-label, and finally zoom in on dataset-specific observations. For the in-domain evaluation, we focus on unique contributing effects per predictive uncertainty method and the relation between method combinations and evaluation metrics.

- When evaluating with proper scoring rules, does an absolute increase in combination size (higher T or M) correlate with better performance?
- What effect—equal over all tasks, datasets or architectures—can be discerned per unique predictive uncertainty method?

2) CROSS-DOMAIN SETTING

Since we test over sentiment classification datasets from multiple domains (Amazon Product Reviews), we seek to analyze uncertainty reliability across domains. However, learned knowledge from a source domain can often transfer to classification in the target domain. Provided this setting we need to account for cross-domain generalization next to out-of-domain detection, the latter which is the focus of our experiments.

Cross-domain generalization - *how well does a classifier trained in a source domain perform on a dissimilar target domain sharing a similar task?* The aim of cross-domain generalization is to learn a robust classifier, which can perform well in multiple domains even if there is limited labeled data in some of the domains. Domain discrepancy is a major challenge, where, for instance, linguistic sentiment expressions used in one domain can be different from that of the source domain. For example, “garbage

disposal” is neutral in kitchen appliances whereas a “garbage movie” is strictly negative. This domain discrepancy challenge is often approached by adaptation [21], [125] or encouraging domain-agnostic feature representations [20], [112]. We propose to test out-of-domain detection with predictive uncertainty as a viable fallback strategy when achieving generalization over domains is difficult.

Out-of-domain detection - *how reliably can a classifier trained in a source domain communicate uncertainty in a target domain provided good/bad generalization?* Whenever a model does not generalize to OOD examples, we would expect a model to be uncertain, allowing detection in order to abstain or trigger conservative fallback strategies [126]. As a proxy to good/bad generalization we measure the gap between in-domain and target domain accuracy as evidence of train-test skew. We argue that our current setting is more realistic than benchmarking OOD detection in totally disparate domains such as evaluating a newswire classifier on movie reviews.

Our analysis will be centered on the following question:

- How does domain similarity affect out-of-domain detection with uncertainty methods? Is there a clear increase of uncertainty given a higher OOD generalization gap?

3) NOVELTY DETECTION SETTING

Novelty detection - *how well can the model identify and communicate uncertainty on samples of a novel class?* In the worst case, classifiers “fail silently” and wrongly attribute high confidence to an in-distribution class [127], [128]. In the best case, the model either lowers its confidence or signals uncertainty. Prior work hypothesizes model uncertainty to be mostly impacted [72], [129].

With this experiment we simulate the conditions of novel class data by removing a single or multiple classes during training. The resulting distribution shift is not too far from the original domain and cannot be considered fully out-of-distribution (as detailed in Subsection II-E).

We determine diverse novelty detection strategies adapted per dataset. For `20news`, we follow [12], [106] and take out all odd-numbered classes to simulate novel distribution shift. Since `imdb` is a sentiment classification dataset, we isolate the middle class, rating “5” out of the 10 ratings, from training and expect the models to allocate prediction mass to a label close to the holdout class (ratings “4” or “6”). `CLINC-OOS` provides a separate out-of-scope intents set on which we assess novel class robustness.

We devise a new strategy for the multi-label classification datasets, where we would isolate a class that is very distinct from the remaining classes, i.e., (i) by not appearing often in the originally multi-label annotated dataset jointly with the remaining classes, and (ii) occurring frequently enough to guarantee representative results. We draw statistics on the label co-occurrence rates of each dataset, and find that for `Reuters` “Acquisitions” (id:0) occurs in 94% of documents as a single topic, making it an ideal candidate for testing novel class detection. For `AAPD` we apply the similar strategy

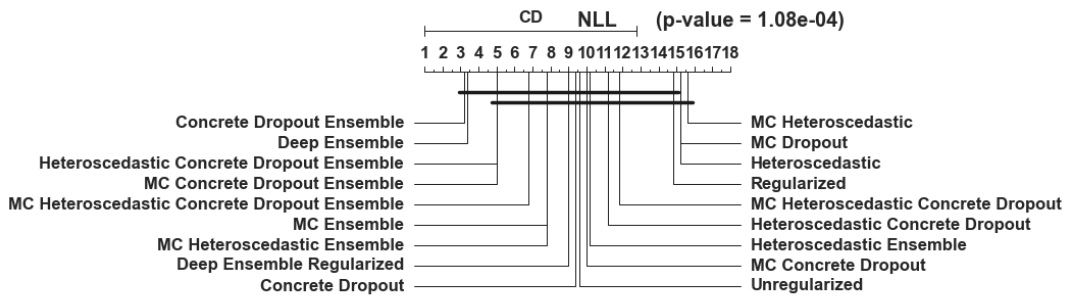


FIGURE 3. *In-domain* results with critical difference diagram comparing all methods by average rank, with the calculated critical difference in the top-left and Friedman χ^2 p-value top-right. *Concrete Dropout Ensemble* achieves the highest NLL rank. While comparing over 5 datasets, the critical difference is large, with only the two aforementioned methods significantly differing from MC Dropout.

and find the frequent label “CS.it” (id:0) to have relatively low label- co-occurrence (2.49), even when there are at least 2 labels to be predicted per sample. We isolate all examples where the novel class appears, either alone or in combination with other labels.

We focus our analysis around three specific questions concerning predictive uncertainty under distribution shift, and compare generally to other modality benchmarks:

- Do hybrid predictive uncertainty methods incrementally or critically improve detection of unseen class instances?
- Does calibration in the in-domain setting translate to calibration under distribution shift?
- Do we see the same trends as in benchmarks from different modalities (Subsection I-A)?

IV. RESULTS

We will present the experimental results in a step-wise manner to avoid confusion on the conclusions to be drawn. We start with general and task-specific trends observed for the in-domain setting, followed by the distribution shift experiments, cross-domain classification and novelty detection. Finally, we present 4 ablation studies on critical, learned or empirically set hyperparameter values. In addition to the visualizations and analyses presented in this main Section, we include the raw evaluation data in Appendix D-B for comparisons and reproducibility.

A. EXPERIMENT: IN-DOMAIN

Naively combining predictive uncertainty methods will not give any absolute performance increase, as proper scoring rules show no correlation (-0.01) with the absolute number of predictive uncertainty methods combined. This requires deeper analysis to identify which singular or hybrid methods do significantly outperform baselines.

First, we visualize **general results** with critical difference diagrams comparing all methods by average ranking over datasets (Fig. 3). *Critical difference* (CD) can be interpreted as the smallest difference between methods which is likely to indicate a significant improvement. In short, the null hypothesis —there is a significant difference between the

TABLE 3. *In-domain* (left) combined *Brier* and *NLL* proper scoring rule pairwise comparison counts of wins/draws/losses and (right) *ECE* metric reported for comparing in-domain calibration. For in-domain predictive accuracy, ensembles clearly are superior. Considering only miscalibration, *Concrete Dropout* generally adds calibration to predicted probabilities. The combination with MC Dropout gives unpredictable ranking results.

ref	wins	draws	losses	ref	wins	draws	losses
9 Deep Ensemble	142	0	28	5 Concrete Dropout	68	1	16
12 Concrete Dropout Ensemble	135	1	34	12 Concrete Dropout Ensemble	58	1	26
16 Heteroscedastic Concrete Dropout Ensemble	130	4	36	4 MC Heteroscedastic	52	1	32
15 MC Heteroscedastic Ensemble	114	2	54	8 MC Heteroscedastic Concrete Dropout	52	0	33
17 MC Heteroscedastic Concrete Dropout Ensemble	114	2	54	2 MC Dropout	49	2	34
11 MC Ensemble	111	3	56	15 MC Heteroscedastic Ensemble	48	1	36
13 MC Concrete Dropout Ensemble	102	0	68	16 Heteroscedastic Concrete Dropout Ensemble	48	0	37
10 Deep Ensemble Regularized	90	1	79	7 Heteroscedastic Concrete Dropout	46	0	39
14 Heteroscedastic Ensemble	82	2	86	9 Deep Ensemble	45	1	39
9 Unregularized	79	4	87	0 Unregularized	40	2	43
5 Concrete Dropout	77	1	92	6 MC Concrete Dropout	40	0	45
7 Heteroscedastic Concrete Dropout	70	3	97	11 MC Ensemble	38	2	45
8 MC Heteroscedastic Concrete Dropout	65	2	103	17 MC Heteroscedastic Concrete Dropout Ensemble	37	1	47
6 MC Concrete Dropout	58	0	112	1 Regularized	32	0	53
4 MC Heteroscedastic	40	5	125	3 Heteroscedastic	29	2	54
2 MC Dropout	39	6	125	14 Heteroscedastic Ensemble	27	2	56
1 Regularized	34	0	136	10 Deep Ensemble Regularized	24	2	59
3 Heteroscedastic	30	0	140	13 MC Concrete Dropout Ensemble	23	0	62

methods— cannot be rejected for all methods connected by a dark bar. We also report *Friedman* χ^2 , which is a non-parametric statistical test that considers ranking methods over different attempts, in our case datasets, requiring a minimum of 3 methods in comparison. This test checks whether the measured average ranks are significantly different from the mean rank that is expected under the null-hypothesis.

Table 3 shows more detailed pairwise comparison scores, demonstrating that if both proper scoring rules are considered, plain ensembles and hybrid methods based on deep ensembles are overall superior to single model uncertainty prediction methods. However, the benefit resides more in accuracy than calibration, where some single model predictive uncertainty methods rank higher, specifically *Concrete Dropout*.

For a most complete answer to unique effects per predictive uncertainty method, we need to analyze **dataset-specific results**. Detailed results per dataset and metrics (Appendix D-A Fig. 22) reconfirm that a method’s superiority (i.e., for the whole application domain of in-domain text classification) should not be concluded based on 1 single dataset. Each dataset has specific problem characteristics, which affect method ranking differently at varying magnitudes. However, the comparative performance of each method is not fully dependent on the dataset tested, with Deep Ensemble performing reliably in-domain as evidenced by rank.

B. EXPERIMENT: CROSS-DOMAIN

This Subsection is dedicated to analyzing predictive uncertainty methods under domain shift. We first present results on cross-domain generalization, followed by a challenging OOD detection setting. Finally, we draw parallels between both settings' experimental results.

We conduct extensive experiments on the benchmark Amazon product review datasets on a total of 12 source-target domain configurations. Each domain is abbreviated by its first uppercase letter: (B)ooks, (D)VD, (E)lectronics, (K)itchen. Fig. 4 reports on the lowest **cross-domain generalization** gap between ID and OOD domain datasets. We observe higher ID accuracy for Kitchen and Electronics, which can indicate a relatively lower complexity of domain sentiment. Importantly, the gap between Kitchen - Electronics and Books - DVD are smallest overall, coinciding with our intuitions on domain similarity. Remarkably, *regularized Deep Ensemble* trained on Book reviews even scores higher accuracy (+1.8%) on its target domain (B→D).

To analyze the cross-domain performance of predictive uncertainty methods we report average rank ID NLL and OOD accuracy (Fig. 5). *Heteroscedastic Concrete Dropout Ensemble* ranks highest in-domain when evaluated with a proper scoring rule. Models without any regularization achieve higher OOD accuracy scores, with *Deep Ensemble* significantly outperforming more than half of the predictive uncertainty methods (first black bar). A possible explanation could be that most target domain data is more similar to the source domain than expected, effectively giving an edge to methods that achieve high ID accuracy.

To evaluate **Out-of-domain detection**, we report AUROC ranks in Fig. 6 and additionally plot OOD detection over generalization scores in Fig. 7. *Concrete Dropout Ensemble* and variations outrank other methods on OOD detection. Nevertheless, we must nuance the ranking results since the magnitude of AUROC is generally low, close to random (50-54%) with no class imbalance, over all 12 cross-domain settings. These results might indicate that from the perspective of the methods tested, there are no salient differences between the different domains. More specifically, Books and DVD as a source have AUROC scores on target OOD domain data centered around 51% and Kitchen and Electronics as a source have comparable AUROC scores with 1 higher AUROC (54%) cluster for OOD Books and DVD targets.

Additionally, Fig. 23 in Appendix D-A demonstrates a similarly clear difference in correlation effect size of uncertainty quantities with ID-OOD data depending on the target domain, e.g., high overall mean correlation (0.3) for Kitchen source evaluated on the disparate domain of Books, whereas uncertainty correlation on Electronics averages around 0.1 for the most correlated quantities.

C. EXPERIMENT: NOVELTY DETECTION

Before analyzing which predictive uncertainty methods provide better detection of instances of an unseen class, we report

on how uncertainty metrics (cf. Subsection II-C4) correlate with novel class data.

In Fig. 8 the final rank over datasets confirms the superior robustness of predictive entropy as an uncertainty metric. Logically, it is closely followed by maximum softmax score. Next, model uncertainty correlates generally well with novel class data. Interestingly, model uncertainty outperforms entropy on AAPD, with most methods showing the need for learning from more data to better approximate the model parameters.

Similarly to the evaluation of in-domain performance, we use CD diagrams (Fig. 10) with binary detection metrics *AUPR* and *AUROC* to provide a ranking of predictive uncertainty methods over datasets.

The absolute pairwise comparisons (Table 9) confirm that hybrid predictive uncertainty methods improve detection of novel class data. Quite surprisingly, *Deep Ensemble* which ranked absolute highest for in-domain, drops multiple ranks in favour of combination ensembles (*Heteroscedastic Ensemble* or even *MC Concrete Dropout*). The in-domain calibration effect from *Concrete Dropout* appears to pass over to this novelty detection setting. More importantly, it also helps boost the novelty detection performance of *Deep Ensembles* when jointly used (e.g., *MC Concrete Dropout Ensemble*).

While comparing over 5 datasets, there is no critical difference between the average ranking of methods, which can point to task or dataset-specific interactions. Fig. 11 shows the variation of AUROC performance for the different methods, from which we can observe that (non-finetuned) dropout sampling (*MC Dropout*) under-performs in most datasets, most clearly on AAPD, by severely underestimating uncertainty on samples of a novel class. We also observe relative benefits of the *Heteroscedastic* loss function for multi-class text classification, which most clearly is represented in the CLINC150 results. The same visualization allows us to evaluate the quality of uncertainty quantification for each method. Generally, epistemic uncertainty derived from ensembles offers higher quality detection of novel class data than single model predictive uncertainty. This effect is clearly visible for multi-class classification where the ensembles clearly group on top, as opposed to the results for the multi-label datasets.

Additionally, we visually detail in Appendix D-A Fig. 24 density estimates for uncertainty quantities with respect to in-domain versus novel data with most hybrid ensemble methods demonstrating better separable densities.

D. EXPERIMENT: ABLATIONS

In this Subsection, we zoom in on the best performing uncertainty prediction methods relative to the complementary benefits hypothesized for hybrid approaches (Subsection III-A), provide explanations for results specific to an architecture (TextCNN vs. BERT, Subsection III-C3), and present ablations on critical hyperparameters.

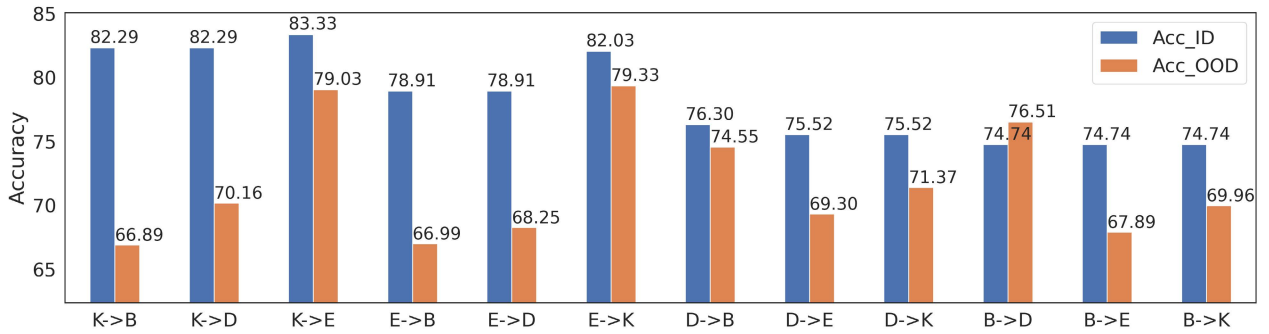


FIGURE 4. Lowest accuracy generalization gap, in-domain (Acc_ID) minus out of domain (Acc_OOD) accuracy (y-axis), of all predictive uncertainty methods per source→target domain combination (x-axis).

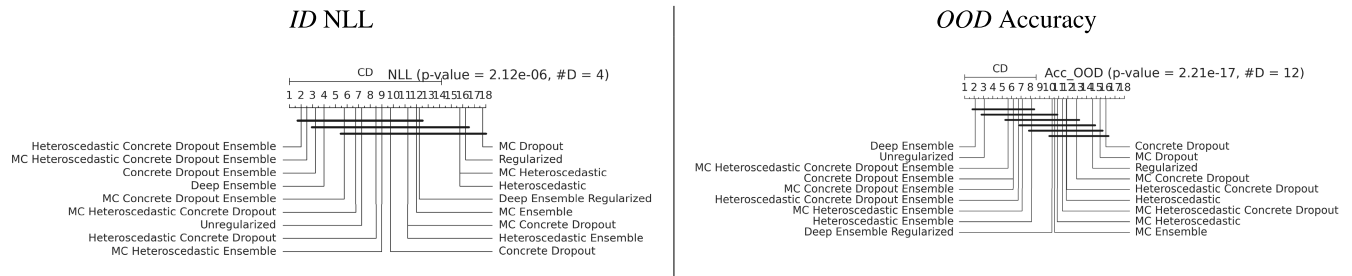


FIGURE 5. Average rank of in-domain NLL for the 4 source datasets (left) and out-of-domain accuracy over 12 source-target configurations (right) for all tested predictive uncertainty methods.

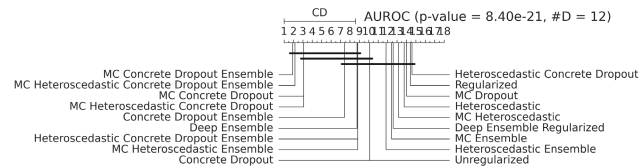


FIGURE 6. Average rank of OOD AUROC over 12 cross-domain settings for predictive uncertainty methods.

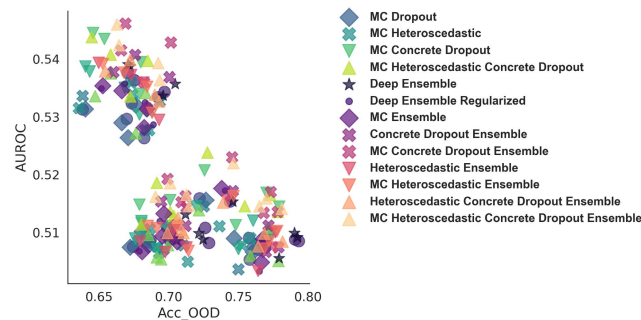


FIGURE 7. AUROC detection magnitude (y-axis) mapped over OOD accuracy (x-axis) with a legend on the right for methods that support uncertainty estimation.

1) DIVERSITY

Diversity of samples drawn from a posterior, either via T MC samples and/or M ensemble components, is an important condition for efficient uncertainty estimation. If each sample

presents a similar function, the overall prediction can be overconfident, and increasingly drawing samples will not reduce this. We derive a small experimental setting from [79] to measure function-space diversity for all predictive uncertainty methods involving posterior sampling.

In Fig. 12 we analyze the relation between accuracy and diversity as measured by Kullback-Leibler divergence between a sampled prediction and the predictive mean, $\frac{1}{T} \sum_{t=1}^T \text{KL}(p(y^*|x^*, \hat{\theta}_t) || \bar{p}(y^*|x^*, \hat{\theta}))$. For a fair comparison, we calculate diversity at the ensemble level if a predictive uncertainty method consists of multiple models, else at the dropout sample level.

While the diversity-accuracy plane does not provide a one-on-one linear relationship, we note in Fig. 12 (a,b,d) promising results for hybrid ensemble methods, which with higher diversity improve on accuracy over Deep Ensemble. The visual of imdb (c) registers overall low diversity, even for simple predictive uncertainty methods which generally achieve higher diversity, albeit by capturing multiple dissimilar yet weaker functions. For AAPD (e), most methods are tied for exact accuracy even with different diversities.

2) NLP ARCHITECTURE

We selected specific representative predictive uncertainty methods on the basis of our previous experiments to run with the Transformer BERT as base architecture. We argue that the chosen architecture can have a non-negligible impact on uncertainty estimation, and we compare with the simple

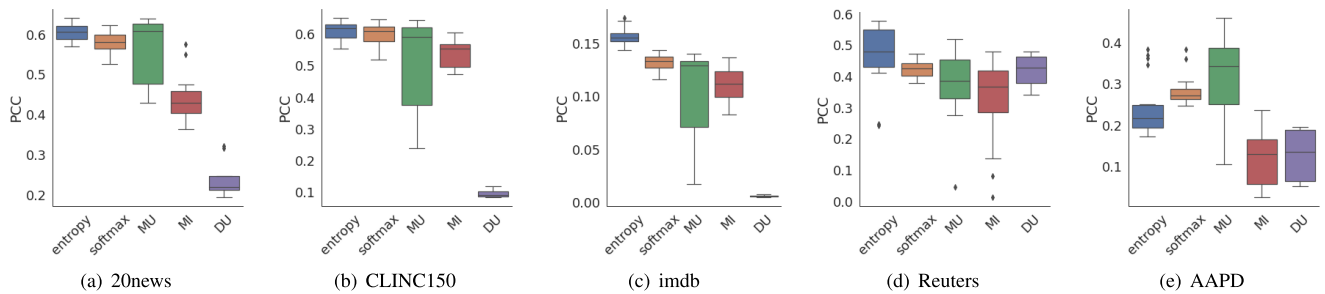


FIGURE 8. We report the Pearson Correlation Coefficient (PCC) between uncertainty values and binary variable ID-OOD for 5 benchmark datasets. Higher absolute correlation score points to stronger association of uncertainty and novelty detection. *Model Uncertainty (MU), Data Uncertainty (DU), Mutual Information (MI).

ref	wins	draws	losses
MC Concrete Dropout Ensemble	121	1	48
Heteroscedastic Ensemble	119	1	50
MC Concrete Dropout	109	1	60
MC Heteroscedastic Ensemble	102	0	68
Deep Ensemble Regularized	100	0	70
Concrete Dropout	90	1	79
MC Heteroscedastic Concrete Dropout Ensemble	89	2	79
MC Heteroscedastic Concrete Dropout	86	1	83
Concrete Dropout Ensemble	83	0	87
Regularized	81	1	88
Heteroscedastic	80	0	90
Deep Ensemble	80	0	90
Heteroscedastic Concrete Dropout Ensemble	75	2	93
MC Heteroscedastic	75	0	95
MC Ensemble	71	2	97
Unregularized	69	0	101
Heteroscedastic Concrete Dropout	47	1	122
MC Dropout	46	1	123

FIGURE 9. Novelty detection CD diagram of AUROC.

yet controllable TextCNN architecture in order to investigate whether the same conclusions hold for novelty detection.

The separate Out-of-Scope set of CLINC150 allows us to easily evaluate novelty detection with BERT. We observe in Fig. 14 on CLINC150 that BERT does increase novelty detection over all metrics. Even without any hyperparameter tuning *Unregularized* BERT outperforms all TextCNN models. Overall, we register the same ranking of predictive uncertainty methods, albeit a Deep Ensemble with BERT is superior to hybrid ensembles. Crucially, we note that the correlation of epistemic uncertainty with novelty detection is higher for each TextCNN ensemble than for every single BERT model.

Most notably, results on all other datasets are inconsistent with the above. For comparison, we have trained an informed sub-selection of predictive uncertainty methods with BERT as base architecture (Fig. 13).

Generally, we observe in (a,b) higher ID accuracy for BERT with relatively slighter gains when ensembling. AUROC scores (c,d) are well below even single TextCNN models, pointing to a crucial deficiency with BERT in a novelty detection setting. The correlation of epistemic uncertainty with novel class samples draws a similar picture (e,f). *MC Heteroscedastic Concrete Dropout Ensemble* on *imdb* does produce more correlated epistemic uncertainty than all other methods.

3) ENSEMBLE SIZE M

Combining models to an ensemble generally benefits performance both in and out-of-domain. Previous research [55], [79] worked out that ensembling benefits stagnate with larger model size M . Fig. 15 selectively reports novelty detection metrics or uncertainty correlation scores for all ensemble-based methods of different sizes.

AUROC score for CLINC150 (15b) is a representative example of the expected effect of ensembling. Importantly, it provides crucial evidence for our general hypothesis, demonstrating that ensembling over predictive uncertainty methods gives complementary benefits in novelty detection settings. What is similarly interesting is that the relative benefit of ensembling shows slightly different curves in certain cases. Epistemic uncertainty for *imdb* (15c) already attains similar performance at $M=2$, again showing comparatively slower (since less required) increase at larger M for hybrid ensembles. AAPD (15e) shows more stagnant behavior for the reliability of entropy with growing ensemble size, irrespective of the predictive uncertainty method.

4) CONCRETE DROPOUT p

Fig. 17 relays an important observation on the dataset-wise adaptation of Concrete Dropout: increasing the learned dropout rate as is required for the problem at hand. This reinforces the argument against fixed-rate dropout.

Reference [70] remarked that practitioners started to adopt the strategy of fine-tuning dropout with a bottleneck pattern, i.e., start with a higher dropout rate in early layers and decrease the deeper you go in the network. Our results (Fig. 16) shows discrepancy with this practice, specifically for *20news* and *CLINC150*. We do note that both converged to low dropout rates, which can provide the basis for this differing behavior.

V. DISCUSSION

Our study investigates both scalable and hybrid procedures for incorporating uncertainty into Deep Learning models for text classification. Next to baseline in-domain uncertainty evaluation, we have designed two experimental settings, novelty detection and cross-domain classification, to analyze the reliability of uncertainty. Additionally, we devised ablation

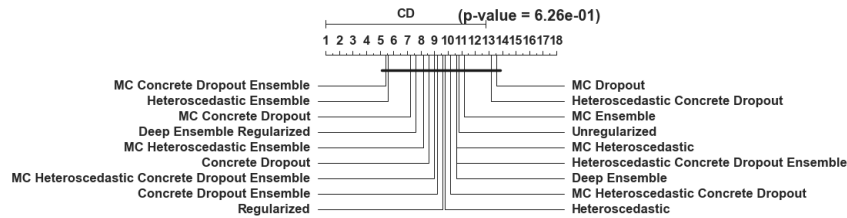


FIGURE 10. Novelty detection CD diagram of AUROC.

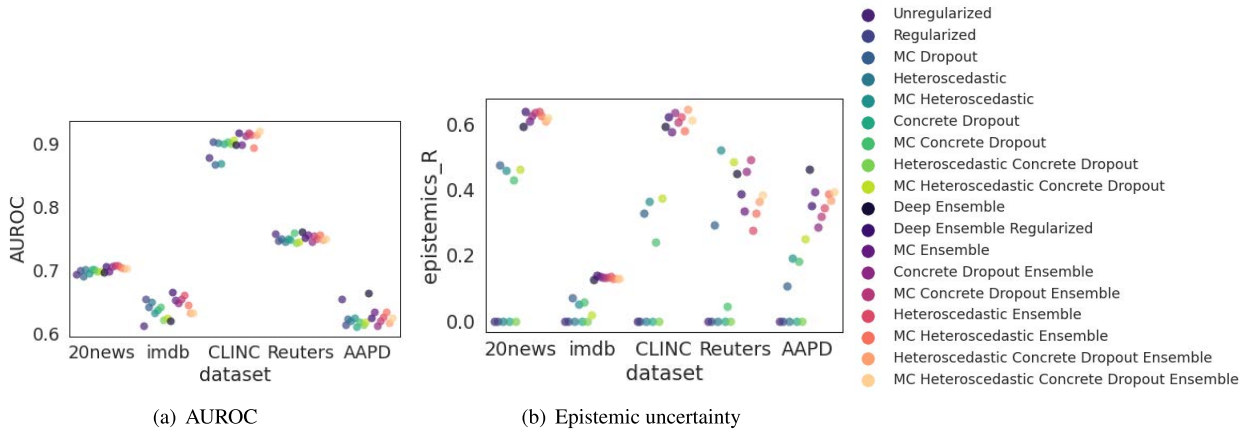


FIGURE 11. Comparison with AUROC(↑) and Epistemic uncertainty PCC(↑) for task and dataset-specific differences in novel class detection. Methods with 0 correlation do not support model uncertainty quantification.

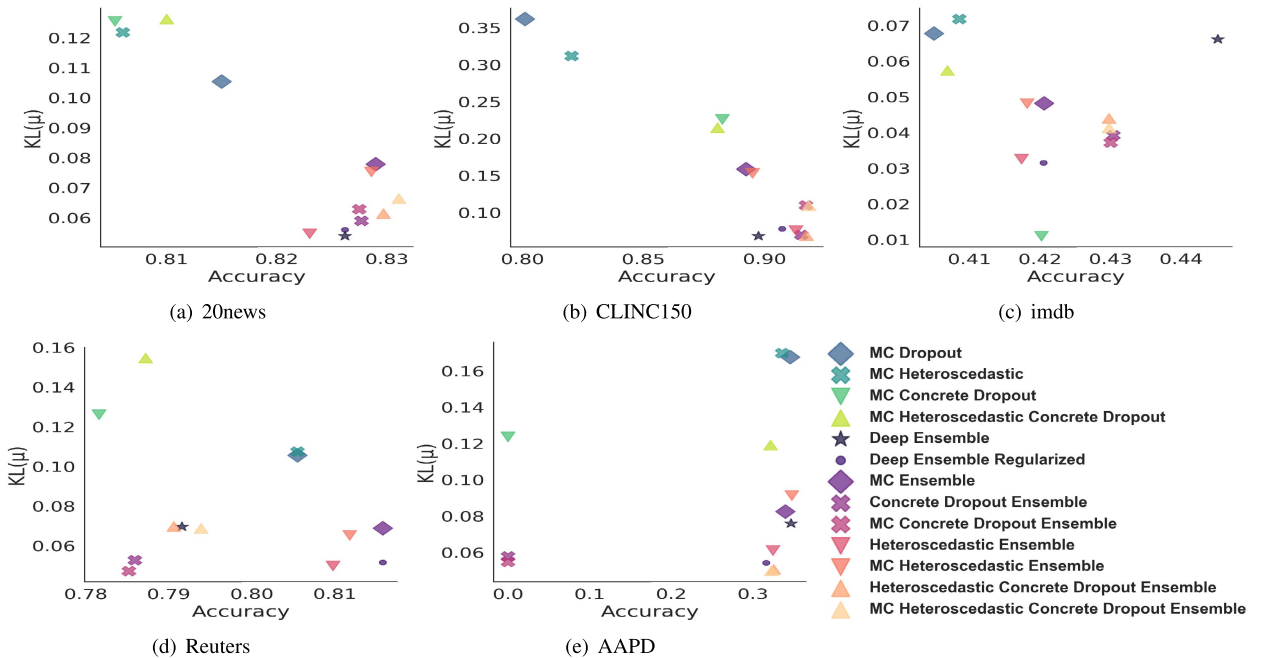


FIGURE 12. Detailed accuracy scores mapped over diversity measured by KL divergence for each of the benchmark datasets.

studies to analyze important hyperparameters in connection to our three hypotheses (Subsection III-A) on complementary benefits for hybrid uncertainty prediction methods.

Benchmarking uncertainty methods We summarize our findings succinctly and discuss the results of each experimental setting.

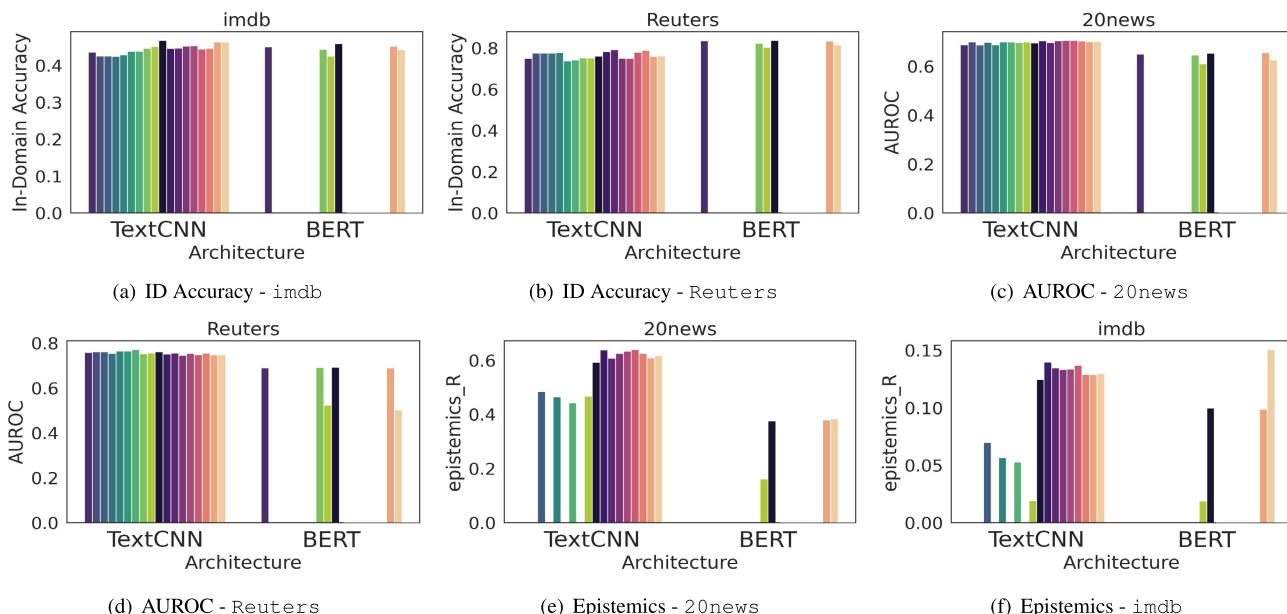


FIGURE 13. Novelty detection scores mapped per architecture for the benchmark datasets without dedicated OOD split. The legend of Fig. 11 applies here.

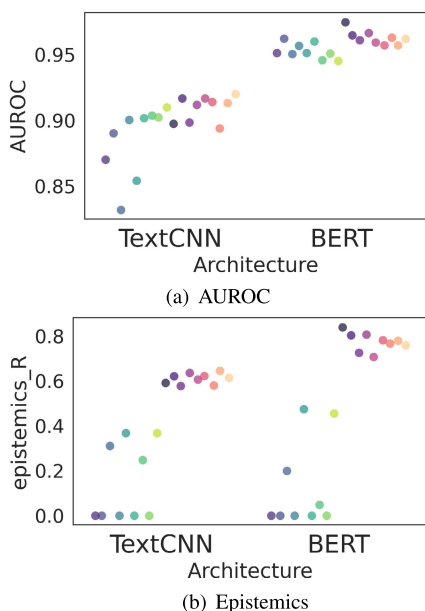


FIGURE 14. Detailed AUROC-epistemics (PCC) scores mapped per architecture on CLINC150. Best performance: upper-right corner. The legend of Fig. 11 applies here.

We find that individually ($>$ indicating “outperforms” over all experiment settings):

Deep Ensemble $>$ *Concrete Dropout* $>$ *(MC) Heteroscedastic* \geq *MC Dropout*

We find that jointly, by considering method combinations: *(MC) Concrete Dropout Ensemble* \geq *(MC) Heteroscedastic Ensemble* $>$ *MC Concrete Dropout* $>$ *Deep Ensemble* $>$ *Deep Ensemble Regularized* $>$ *MC Dropout*

In-domain results (Subsection IV-A) corroborate the superiority of *Deep Ensemble* with high accuracy and proper scores (NLL, Brier). *Table 3* demonstrates that the improvements come from accuracy as opposed to calibration, where *Concrete Dropout*-based methods rule.

Cross-domain experiments (Subsection IV-B) give differing conclusions: cross-domain generalization results are similar to in-domain, whereas out-of-domain detection follows novelty detection results. Our evaluation of uncertainty quantities (*Fig. 23*) demonstrate reliably higher correlation of uncertainty with domain discrepancy. We do take note of relatively low magnitude AUROC (*Fig. 6*), which underlines how challenging out-of-domain detection is in a domain adaptation setting with comparably similar linguistic patterns.

Novelty detection (Subsection IV-C) in text classification gives reverse results: Hybrid ensemble methods with *Concrete Dropout* rank highest scored by AUROC, AUPR and model uncertainty correlation, followed by other method combinations that induce calibration. We do note that specific method performance is often tied to task and dataset characteristics, with results averaged over the 5 benchmark sets showing statistically non-significant differences between methods. As shown in *Table 9*, standard *Deep Ensemble*, i.e., without any regularization or prior from combining methods, perform worse outside the in-domain setting. The case for standard *MC Dropout* is even worse with novel class robustness (AUROC and AUPR) lower than the *Unregularized* point-estimate model.

Remarkably, BERT performs worse than the simpler TextCNN model at detecting distribution shift in the form of novel class data (*Fig. 14*). Results on the OOS set of CLINC150 differ from results obtained on all other datasets,

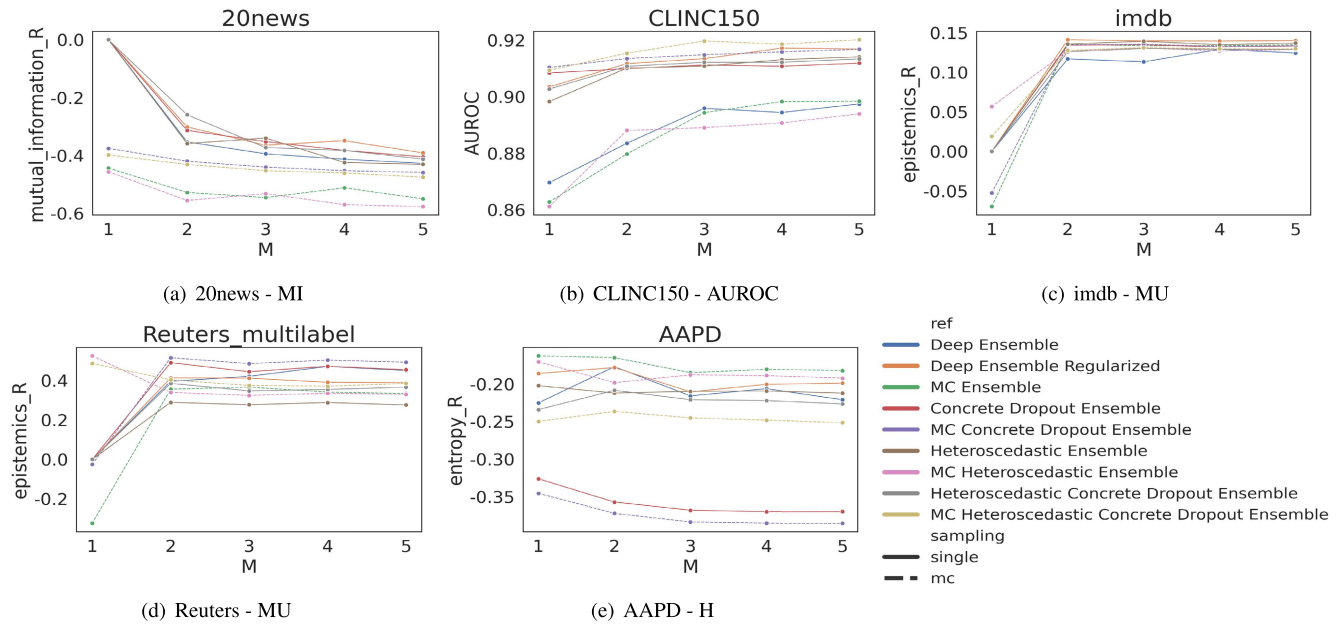


FIGURE 15. Visualization of representative dataset-quantity/metric combinations mapped over stepwise increasing ensemble size M . Note that positive and negative correlations are corollary to the quantity reported. Given the small relative differences, plots are best viewed online.

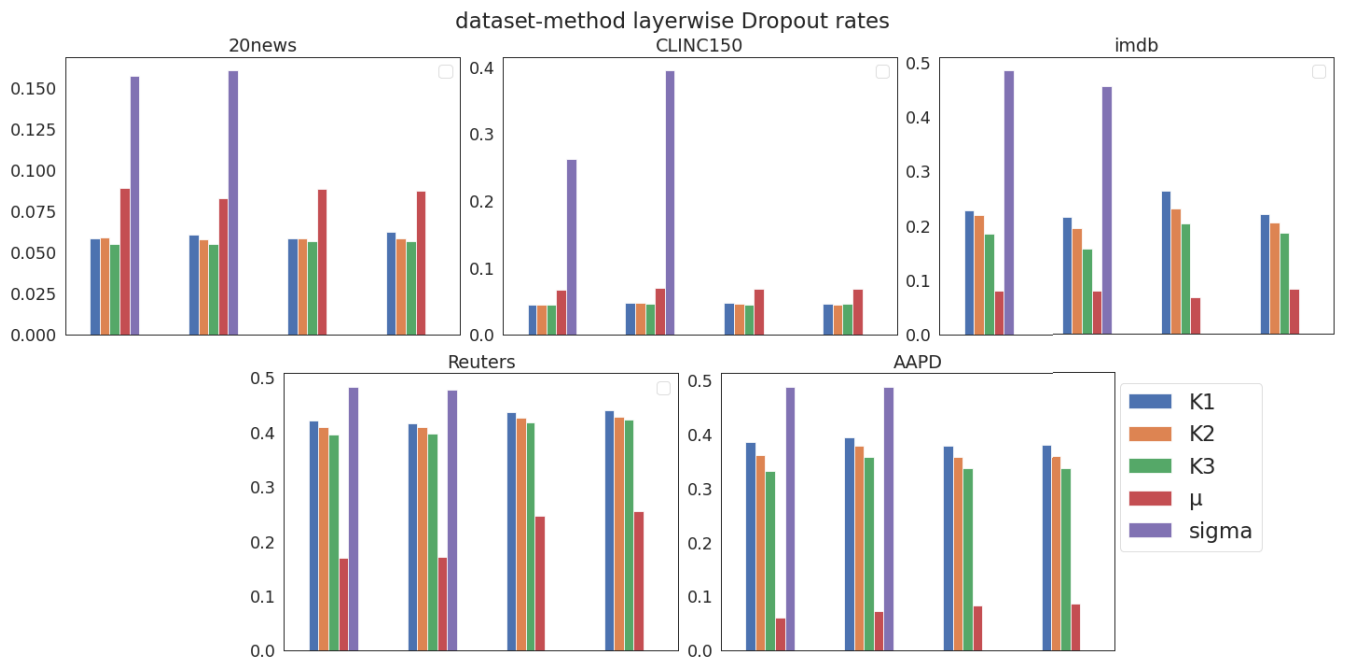


FIGURE 16. Learned layer-wise dropout probability per layer for each method with Concrete Dropout. The first 3 layers are the CNN kernels ($K1 - 3$), followed by the penultimate layer μ , possibly with σ for modeling heteroscedasticity. The legend of Fig. 17 applies here.

which we believe can be attributed to the short, in-domain intent commands differing strongly in vocabulary with the OOS samples, resulting in a comparatively less challenging novelty detection setting. We contend that novelty detection is actually more challenging for BERT despite of its pre-trained language modeling knowledge and because of the strict requirement to fine-tune the task-specific final layer

with new supervision. Its ability to detect (and overly rely on, e.g., [130]) statistically relevant yet possibly spurious cues in language data will make it overconfident with transfer to a new task when the IID assumption cannot be maintained.

Validating hybrid approaches We have empirically analyzed individual-joint effectiveness in modeling predictive uncertainty and will answer our three hypotheses

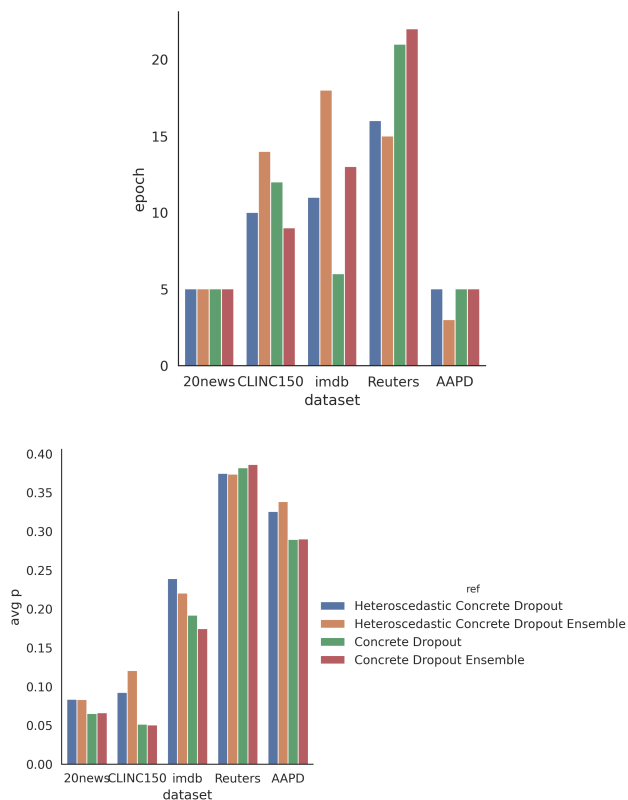


FIGURE 17. Top: Average epoch of convergence per dataset. Bottom: Average learned Concrete Dropout probability per dataset over predictive uncertainty methods. We observe very dataset-dependent dropout rates.

on complementary benefits from combining inter and intra-modal posterior approximation.

Firstly [A], ensembling (increasing M) proves to give relatively higher performance benefits than stochastically sampling predictions from an optimized solution (T). The effect is clearest in the in-domain setting (Table. 3) and is less pronounced in the out-of-domain settings. For a given predictive uncertainty method, we cannot provide solid evidence that uncertainty reliability always improves when subspace sampling (increasing T , “MC”). AUROC and AUPR rankings (Figs. 10 and 6) present evidence in favour, although Fig. 11 depicts a more fine-grained comparison over datasets and uncertainty methods. Our analysis of diversity (Fig. 12) shows promising results for hybrid ensemble methods, which exhibit higher diversity in posterior samples resulting in improved accuracy.

Secondly [B], our newly proposed hybrid uncertainty estimation methods improve effectively over singular methods, both in novelty detection (Table 9 and Figs. 10, 11) and out-of-domain detection (Fig. 6). Additionally, in ablation studies we find (Fig. 15) that combining predictive uncertainty methods in an ensemble attains higher performance with a lower number of models ($M < 5$) compared to a Deep Ensemble ($M = 5$).

Thirdly [C], Table 3 demonstrates that MC Concrete Dropout improves over MC Dropout ($p=0.5$) on ECE and

proper scoring functions. The out-of-domain experiments (detail: Fig. 11) similarly show that not fine-tuning dropout to the dataset and task at hand is detrimental even when combining models into an ensemble (e.g., MC Ensemble vs. MC Concrete Dropout Ensemble). Ablation on Concrete Dropout (Fig. 17) points to very dataset-dependent learned probability rates, which vary strongly layer-wise (Fig. 16). We link the empirical superiority of MC Concrete Dropout Ensemble to balanced posterior collapse, thanks to the VI-based optimization of the dropout prior. We tentatively claim that the former provides constrained hypothesis support and a more fine-tuned influence of prior.

Benchmark comparison When comparing our results to existing BDL benchmarks, most observations are consistent for in-domain and out-of-domain performance.

Our in-domain results are most similar to [12], where Deep Ensemble outperforms most methods, —albeit in their survey they did not compare combinations of predictive uncertainty—, in our benchmark closely followed by hybrid ensemble methods. When evaluating over various data retention rates [11] observed that “an ensemble of MC Dropout models” (our MC Ensemble) consistently outperforms all other methods. This survey offers the closest point of comparison, although our experimental settings vary. While we cannot directly compare cross-domain detection with other benchmarks, we argue that our cross-domain classification setting mimics their low data regime experiments.

Across different modalities and tasks, Deep Ensemble has been reported to consistently outperform VI-based methods, most specifically MC Dropout, with/without distribution shift (image classification [12], molecule prediction [131], and pendulum physics [132]). However, for a binary image classification problem, [11] report higher accuracy for MC Dropout compared to Deep Ensemble, whereas our results suggest that MC Dropout can induce positive calibration, yet score lower on accuracy and with proper scoring rules. In their experiments they use a fixed dropout rate of 0.2 and fine-tuned weight decay rate, making them fitting for their task at hand and explaining possibly optimistic results. Another uncertainty quantification benchmark [15] reports strong results on image classification for various Monte Carlo methods, although we cannot make a direct comparison. For further discussion, we refer the reader to Appendix C-A.

Our results suggest that BERT performs worse in a novelty detection setting, whereas [42] concludes that Transformers are considerably more robust when compared across domains, e.g., detection of news samples with a sentiment classifier. We point out below that both settings are in fact incomparable. We evaluate detection on novel samples which have alike vocabulary characteristics to the source domain albeit they are excluded from training supervision. Their setting evaluates detection between very disparate domains where linguistic patterns are significantly different and BERT will most probably fallback to its pre-trained knowledge for detection. In short, we do believe that pre-trained Transformers could perform better under varying distribution shifts, yet

with our results underpinning the exception of novel class detection. More research is needed into how the inductive bias from given NN architectures influences approximate inference.

Take-homes For predictive uncertainty in text classification, we derive a number of take-homes from the benchmarking evidence, centered around practical facets to consider for applications.

One has to consider (i) ease and cost of implementation, (ii) computational and memory complexity, comprising training compute, test compute and storage/memory constraints, (iii) the degree of fine-tuning required, (iv) type of supervision; multi-class with low/high number of classes (K) or multi-label with low/high cardinality (C), (v) expectation of distribution shift; in the form of novel class data or unseen language patterns, and (vi) support for uncertainty quantification by source.

For a prototypical low K multi-class text classification task, we advise *Deep Ensemble* for solid in-domain performance and adequate distribution shift robustness. In the case of memory or storage constraints, for example if your base model already has high complexity, using (MC) *Concrete Dropout* will provide calibration benefits both in and out-of-domain, albeit at a slightly larger implementation cost. Similarly, to constrain computational complexity, it can be more sensible to rely on a TextCNN ensemble (5*6M parameters) rather than BERT (110M parameters). Considering time complexity, we have added detailed compute, time and storage statistics for evaluated methods (Appendix Subsection B-B). We would advise against using *MC Dropout* if the dropout rate and weight regularization are not fine-tuned for the problem at hand. Our benchmarking experiments demonstrate the unpredictable behavior of fixed-rate *MC Dropout*, compared to *Concrete Dropout*, which we used as a proxy for models with fine-tuned dropout ratio. This (mal)practice should be highlighted as it has substantial impact on uncertainty estimation and robustness.

If K starts to increase, it warrants the effort to implement the *Heteroscedastic* loss function, which will make the model more calibrated in-domain. Additionally, it enables data uncertainty estimation for possible noisy ground truths, which can happen more frequently with a larger number of classes.

If C grows larger, reliable epistemic uncertainty estimation becomes more important, since the problem is made more complex given the larger number of label combinations. Our evidence is slightly contradicting, with results obtained on Reuters suggesting *MC Concrete Dropout Ensemble* and on AAPD warranting *Deep Ensemble*. What should be clear, is that any form of ensembling is valuable in multi-label classification to boost performance.

Under the expectation of distribution shift in the form of novel class data, adding *Concrete Dropout* with stochastic sampling to an ensemble, *MC Concrete Dropout Ensemble*, gives relatively strong benefits compared to a regular

Deep Ensemble. Ablations also show that less models (M) would be required to reach similar performance. Generally, in-domain calibration inducing methods are more robust when applied in the tested out-of-domain settings. For the in-domain setting, the incorporation of data uncertainty incrementally improves multi-class text classification. Ablation on NLP architectures (Subsection IV-D2) points to a deficiency of BERT for detecting novel class data and would similarly be advised against in favour of simpler text classification architectures.

VI. LIMITATIONS

As with the majority of benchmarking literature in Bayesian Deep Learning, the design of the current study is subject to limitations.

The first limitation concerns selection bias for text classification datasets. We benchmark 6 prototypical text classification datasets covering binary, multi-class, and multi-label classification by topic, sentiment and intent. The task domain of text classification is very large with additionally interesting variations of (i) short social media or long business document text, (ii) hierarchical or extreme multi-label text classification, and (iii) challenging task settings such as fake news detection or reading comprehension. Since these present open sub-problems in text classification we did not consider them for our benchmarking study, yet encourage analysis for future research.

The second limitation is related to the representativeness of uncertainty quantification methods. We specifically opted for scalable procedures which have been increasingly gaining attention by practitioners. In total we derive 18 method combinations from two competing predictive uncertainty procedures, for which we already resort to statistical summaries and rank-based evaluation to present results. Due to computational constraints, retraining min. 5 ensembles of size $M = 5$ per dataset and per experiment setup, we did not consider a natural Bayesian extension of Deep Ensemble, *Bayesian Ensemble* [10] where all weight initialization is shared around a single prior. Additionally, in Appendix C we include preliminary experiments with two new uncertainty approaches, *cyclical SG-MCMC* [60] and *SNGP* [63], which are less practical to benchmark, but bring promising ideas for improved, high-quality uncertainty estimation.

Finally, evaluating the quality of uncertainty quantification is an open problem in BDL, typically approached with proxy setups, as is the case in our benchmark with a focus on novelty detection and cross-domain generalization. Subsection II-E presents a nuanced view of this evaluation practice. In addition, evaluating reliable uncertainty estimation in NLP as opposed to other modalities is complicated due to the discrete nature of language. Ideally, we would have extended our benchmark with more probing setups covering situations where we expect predictive uncertainty to be crucial, for instance, when dealing with noisy supervision/inputs or low data regimes.

VII. CONCLUSION

In general, while seeking to optimize for a well-approximated (whether or not Bayesian) posterior, current predictive uncertainty methods are imperfect and very often practically not useful. However, the need for practical and scalable solutions to both incorporating and evaluating the quality of uncertainty is huge, as it is a prerequisite to reliable automation. Uncertainty quantification requires modality to task-specific benchmarking to help practitioners safely rely on them and inform researchers to prioritize the right approaches.

In this work, we have presented empirical evidence from benchmarking uncertainty methods in text classification, contributing and calling attention to the under-explored study of uncertainty quality and model robustness in realistic NLP data distributions.

Interestingly, we find that general behavior of predictive uncertainty methods does not hold over different datasets, with method performance often tied to the text classification task. Overall, we cannot discern a clear winning predictive uncertainty procedure, yet some methods clearly perform worse. Although a universal methodology is absent, we observe that there are specific correlations between a method's performance and the problem setting representing text classification task characteristics, for which we have formulated practical take-homes.

An important contribution is the proposed novel combinations of predictive uncertainty methods. Our benchmarking experiments have revealed *MC Concrete Dropout Ensemble* to be overall superior at novel class and out-of-domain detection in text classification, even with a lower ensemble size. Most notably, it outperforms Deep Ensemble which has leading performance in recent BDL surveys on image data. We linked complementary benefits of hybrid uncertainty estimation methods to ongoing research on NN diversity in function-space and have provided more evidence in support of hybrid approaches. We have determined in an ablation study that M , ensemble size, T , number of Monte Carlo samples, and p , dropout probability rate, are crucial hyperparameters to take into consideration for improved robustness and uncertainty estimation. Finally, we experimentally validated predictive uncertainty methods on real-world text classification tasks, including multi-label targets, coupling our hypotheses and results to the NLP problem space. Crucially, we found an important deficiency of BERT, compared to a more simple NLP architecture TextCNN, with respect to novel class robustness, limiting the applicability of transfer learning from pre-trained Transformers under the expectation of uncertainty and novel class instances.

To further improve calibration and robustness in the text classification domain, and by extension uncertainty in NLP, we need to better understand what will make existing or novel uncertainty estimation techniques successful. This requires the development of well-motivated tooling and protocols to reliably assess the quality and fidelity of posterior approximation. Generally, the role of priors in increasingly larger

TABLE 4. Compute and storage costs in Big-O notation [12] for uncertainty methods.

	Method	Compute/N	Storage
MC (Concrete) Dropout	Baseline	m	m
	Heteroscedastic	mT	m
	Deep Ensemble	$m + l(T - 1)$	$m(+l)$
	cSGMCMC	mT	mT
	SNGP	$m + l^2$	m

models deserves more attention. While our work focused on posterior geometry and weight-based priors in the form of regularization, stronger, more meaningful functional priors exist, which should be exploited to encourage desirable predictive behavior such as robustness to specific distribution shift. Particularly for NLP, more focused research is required into what aspects—language data characteristics, inherent task difficulty or ambiguity, architecture design, learned representations, objectives, and effective parameter usage—render NLP pipelines more complex to imbue with reliable uncertainty and guarantee future out-of-distribution robustness.

APPENDIX A IMPLEMENTATION DETAILS

In this Section, we describe the implementation details for the different datasets, architectures and inference methods used in our benchmark.

A. SOFTWARE AND DATA

We have published our benchmarking software at <https://github.com/Jordy-VL/uncertainty-bench> so that the community can continue to build on our work. We have added detailed instructions for reproducibility and extensibility. This allows anyone to test on a new dataset of interest, implement a new uncertainty estimation method, or evaluate on <https://github.com/Jordy-VL/uncertainty-bench/tree/main/datasets>.

B. HYPERPARAMETER DEFAULTS

For each baseline architecture and uncertainty method combination, we describe hyperparameter values in detail for facilitating future replication.

Our choice of hyperparameter values for TextCNN is heavily based on [114], for fine-tuning BERT on [44], [115] and we draw inspiration from [41] for uncertainty estimation method parameters. We seek to restrict hyperparameters as much as empirically plausible to 1 static setting over datasets per architecture.

We constrain the input vocabulary to the 20,000 most frequent words (30K for *20news* and *AAPD*), retain the original document lengths, remapping tokens with a frequency lower than 3 to *UNK* and *PAD* tokens are masked throughout. For TextCNN 300-D embeddings are uniformly initialized upon which three different kernels (3,4,5) operate with 100 feature maps per kernel followed by a max

TABLE 5. CLINC-OOS models with training timings (in seconds) per epoch and total running time.

methods	architecture	train time/epoch	epoch finished	train runtime
Unregularized	TextCNN	32	8	256
Regularized	TextCNN	32	28	896
Heteroscedastic	TextCNN	59	17	1003
Concrete Dropout	TextCNN	35	12	420
Heteroscedastic Concrete Dropout	TextCNN	58	10	580
Unregularized	BERT	420	5	2100
Regularized	BERT	691	11	7601
Heteroscedastic	BERT	710	16	11360
Concrete Dropout	BERT	679	9	6111
Heteroscedastic Concrete Dropout	BERT	707	16	11312

TABLE 6. CLINC-OOS models with inference timings presented in unit time for how many batches or samples can be processed in 1 second wall-clock time over CPU and GPU. For the short sequences of CLINC, both models allow a batch size of 32.

architecture	method	# batch (gpu)	# sample (gpu)	# batch (cpu)	# sample (cpu)
TextCNN	Unregularized	59.0	1891	63.0	2043
TextCNN	Regularized	66.0	2134	60.0	1922
TextCNN	MC Dropout	53.0	1708	32.0	1050
TextCNN	Heteroscedastic	693.0	22176	482.0	15444
TextCNN	MC Heteroscedastic	47.0	1525	38.0	1216
TextCNN	Concrete Dropout	66.0	2130	40.0	1293
TextCNN	MC Concrete Dropout	48.0	1541	25.0	827
TextCNN	Heteroscedastic Concrete Dropout	756.0	24205	318.0	10197
TextCNN	MC Heteroscedastic Concrete Dropout	48.0	1561	27.0	874
BERT	Unregularized	6.0	223	0.8	25
BERT	Regularized	9.0	306	0.8	26
BERT	MC Dropout	0.9	28	0.1	2
BERT	Heteroscedastic	10.0	325	0.8	26
BERT	MC Heteroscedastic	1.0	31	0.1	2
BERT	Concrete Dropout	7.0	245	0.9	27
BERT	MC Concrete Dropout	1.0	30	0.1	2
BERT	Heteroscedastic Concrete Dropout	6.0	218	0.9	27
BERT	MC Heteroscedastic Concrete Dropout	0.9	30	0.1	2

pooling operation. For BERT we tokenize and encode using the standard BERT tokenizer with maximum sequence length determined per dataset [20news: 250, CLINC: 50, IMDB: 350 and Reuters/AAPD: 200].

Following the MC Dropout procedure we apply dropout [133] with a rate of 0.5 after each non-linear weights layer, which is detailed per architecture in Fig. 2. We found a global weight decay rate of $1e-4$ [134], [135] to work well for TextCNN, whereas we disabled weight decay for BERT since it overpenalized model complexity, resulting in vanishing gradients.

During training TextCNN, Adam optimizes cross-entropy or heteroscedastic loss (see Section II-B4) with a learning rate of $1e-3$ for 45 epochs on batches of size 32. For fine-tuning BERT, we schedule the learning rate starting from $1e-5$ to $1e-6$ with batch size 16 and train for 20 epochs (longer than the original recommendation, following [136]). We use early

stopping conditioned on the validation loss with sufficient epochs to ensure all models are trained until convergence. Else the models might have learned to approximate well the mean of the predictive posterior distribution, but not the variance. At evaluation time, we estimate predictive mean and uncertainties by drawing T samples from the approximated predictive posterior distribution or by averaging over M models. We have empirically set T to 10 and for ensembles the number of models M to 5.

APPENDIX B PRACTICAL CONSIDERATIONS

A. TAKE-HOME SUMMARY

Concretely, for a multi-class problem with a large number of classes, incorporating input-dependent data uncertainty improves accuracy and novelty detection. With high label cardinality in multi-label classification, we recommend

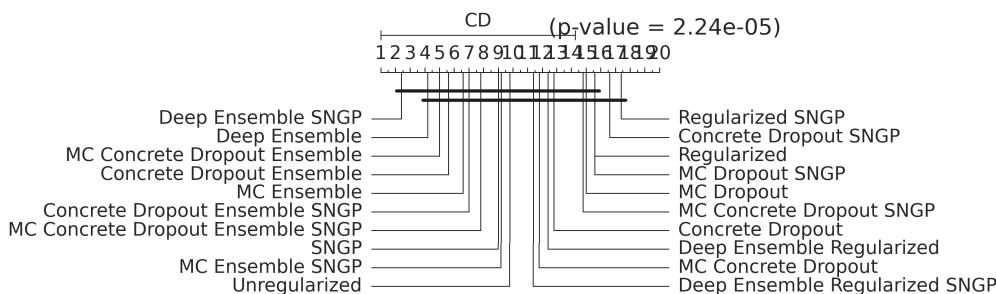


FIGURE 18. CD diagram of NLL for base and SNGP method combinations with a TextCNNv2 backbone.

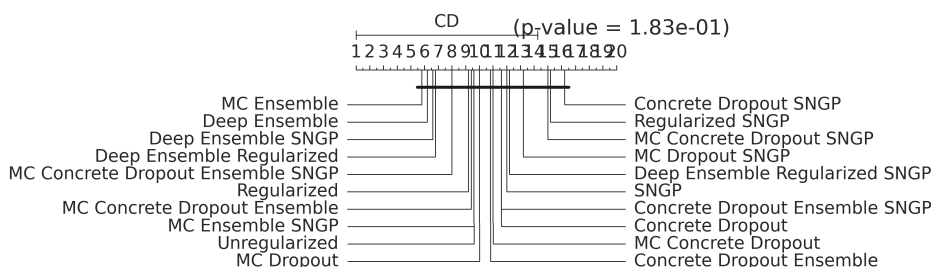


FIGURE 19. CD diagram of AUROC for base and SNGP method combinations with a TextCNNv2 backbone.

ensembling for more reliable epistemic uncertainty estimation. More generally, we advise against using *MC Dropout* if the dropout rate and weight regularization are not fine-tuned for the problem at hand, drawing parallels to dropout probability rates adaptively learned with *Concrete Dropout*.

Hyperparameter considerations We reiterate important hyperparameters and reasonable defaults for text classification tasks similar to our benchmark setup and applications of the above.

- Dropout rate p : the original work suggested a fixed binary rate ($p=0.5$), whereas our experiments indicate different rates are more applicable per dataset. It is best to cross-validate layer-wise dropout probabilities for any real-world application, where impossible it warrants the low effort of incorporating *Concrete Dropout*, consequently reducing experimentation time.
- Weight decay $L2$: best to start with small values [$1e-6 - 1e-4$] and fine-tune accordingly. Take note to not apply global weight decay in case of pre-trained weights, which already have high weight magnitudes, possibly impeding learning.
- MC Dropout T : a small number ($T=10$) of stochastic samples suffices, if large number of classes, scale sub-linearly with K . T also applies to the number of samples drawn to calculate heteroscedastic loss, so beware increasing to too large values since it affects training compute.
- Ensemble size M : a total of ($M=5$) ensemble models is plenty, certainly when combining with fine-tuned dropout rate at the individual model level.

B. COMPUTE VS. PERFORMANCE TRADE-OFF

Next to performance, practitioners are generally concerned with computational and memory costs. [15] present similar concerns in the benchmarking of uncertainty methods. Considering the cost of compute vs. storage, each uncertainty method impacts both differently. Following [12], we present computational and memory costs for evaluated methods symbolically (Big-O), with m flops or storage for a trained model, l represents flops or storage for the last layer, T denotes sampling or replications, and ι GP inducing points.

Our experiments were carried out on a system with a Intel Core i7-10750H 2.6 GHz CPU and NVIDIA GeForce RTX 2070 Max-Q GPU.

Additionally, we provide an informative table with training (Table 5) and test (Table 6) timings provided over all single models on CLINC-OOS.

**APPENDIX C
ADDITIONAL UNCERTAINTY APPROACHES**

Next to the method combinations benchmarked in the main work, we acknowledge two alternative approaches to uncertainty estimation with appealing properties such as training scalability and cheaper inference.

A. STOCHASTIC GRADIENT MCMC METHODS

There exists a wide range of sampling-based inference methods in the stochastic gradient MCMC (SG-MCMC) literature, which have become increasingly more tractable and empirically successful for uncertainty estimation.

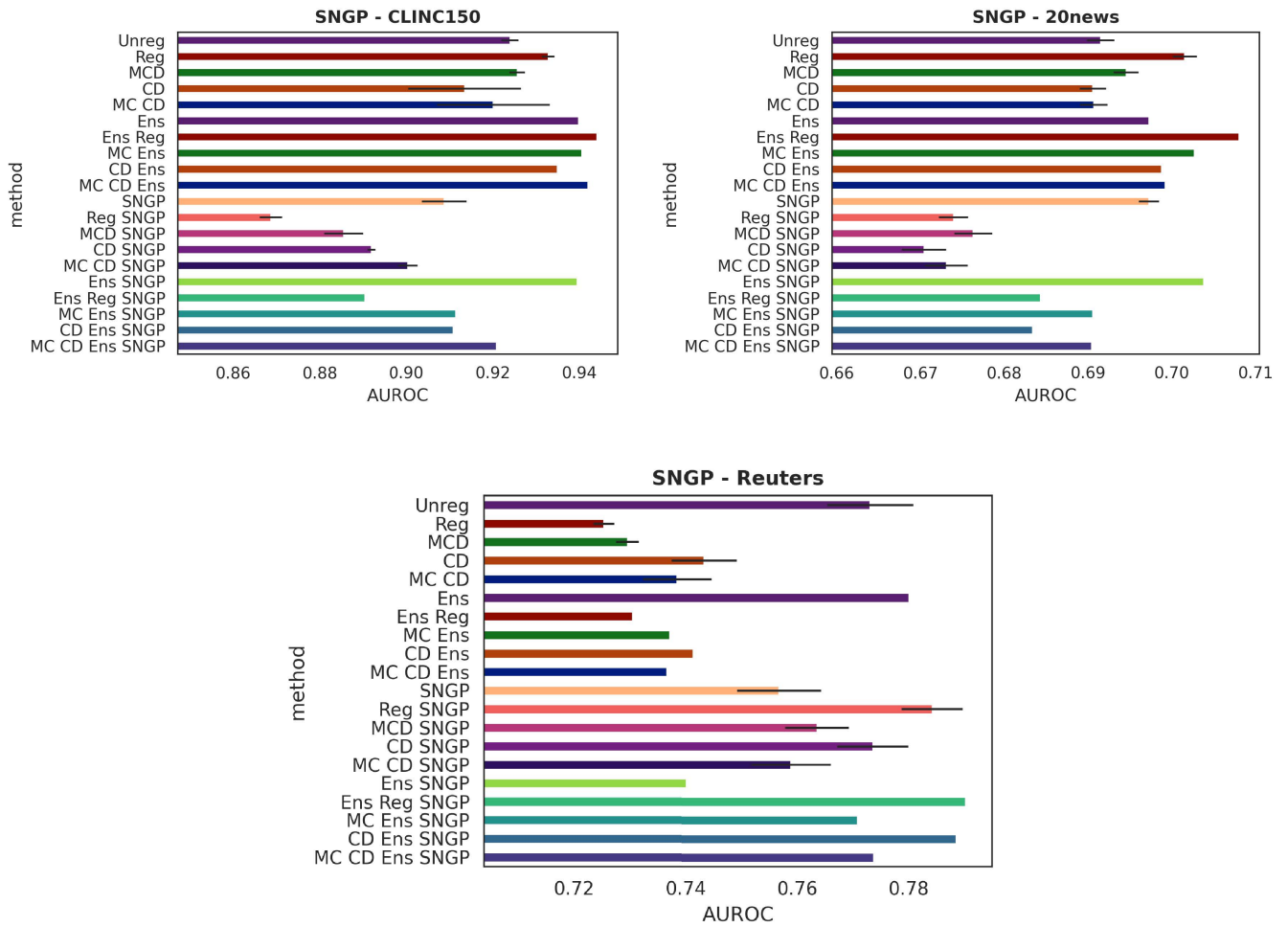


FIGURE 20. AUROC scores over unique (abbreviated) methods per dataset. Error bars are computed over multiple runs (5 seeds) for non-ensembles.

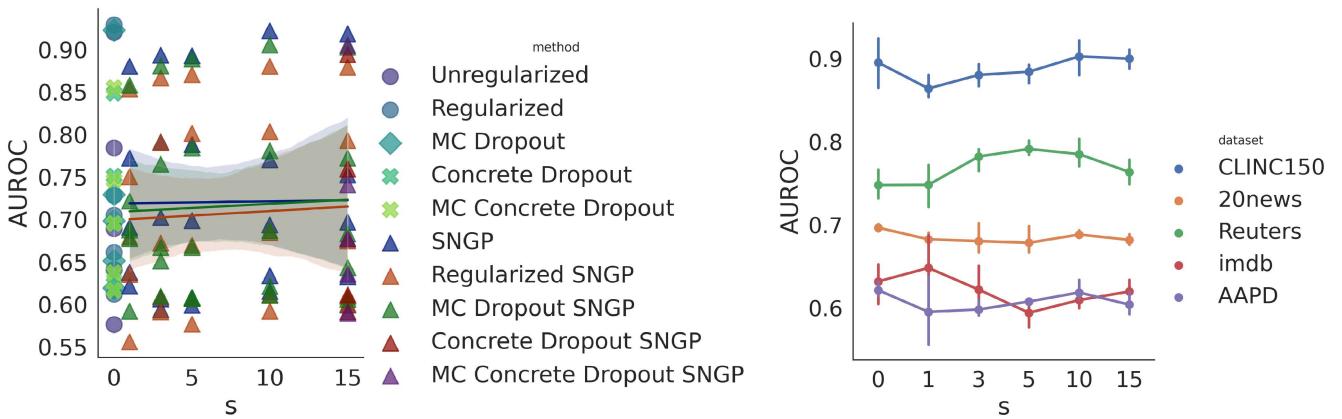


FIGURE 21. Left: AUROC scores (y-axis) over all datasets with unique runs plotted for base ($s = 0$) and SNGP TextCNNv2 models with varying spectral normalization multipliers (x-axis). Lines with shading indicate the trend observed between AUROC and s . Right: AUROC mean and stddev over runs, sampling and datasets.

Specifically, we re-implemented an exemplary approach [60], *cyclical SG-MCMC* (cSG-MCMC), which uses a cosine cyclical learning rate schedule [137] to (i) better explore

the highly multimodal loss landscape and (ii) sample more efficiently from the posterior. While this appealing approach reduces computational complexity by only training a single

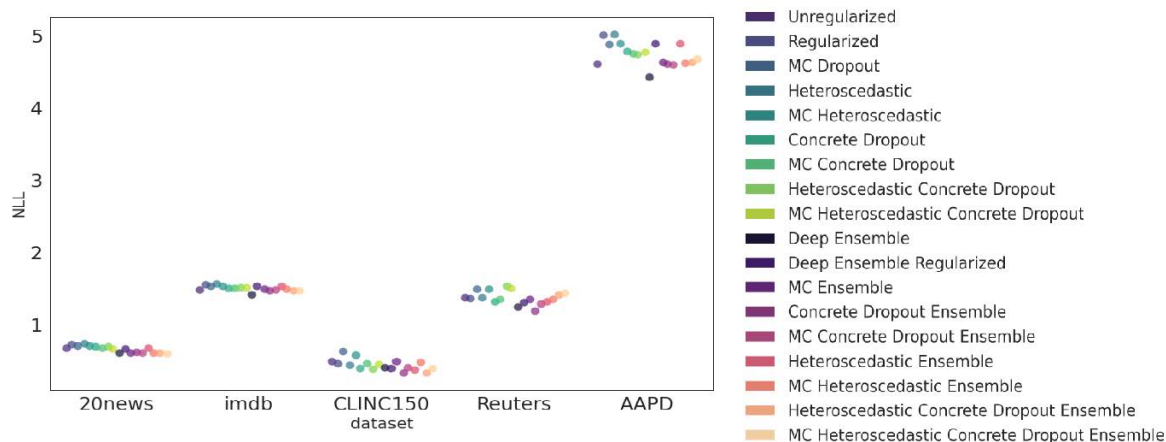


FIGURE 22. Comparison with NLL(↓) for dataset-specific differences in method performance.

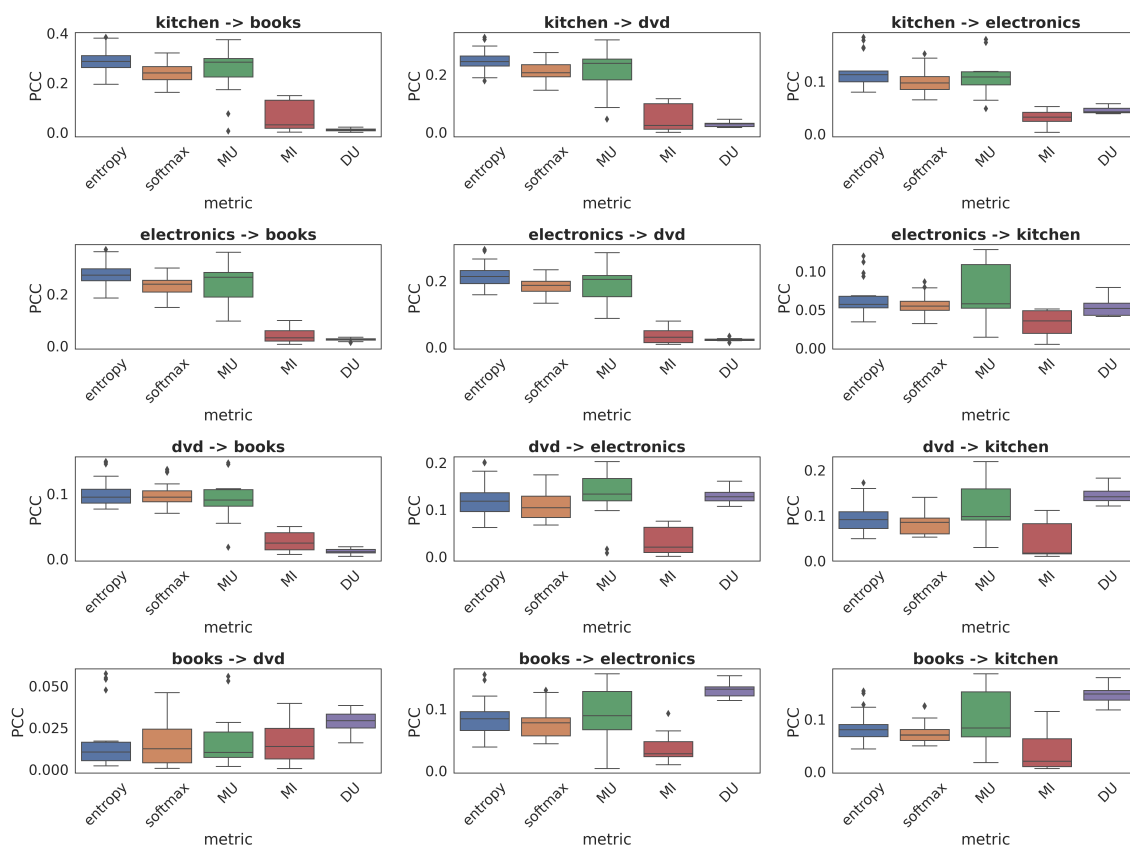


FIGURE 23. We report the Pearson Correlation Coefficient (PCC) between uncertainty values and binary variable ID-OOD for Amazon product review datasets. A higher absolute correlation score points to stronger association of uncertainty and out-of-domain detection. *Model Uncertainty (MU), Data Uncertainty (DU), Mutual Information (MI).

model, we experienced that it is very tricky to finetune with many hyperparameters interplaying. Instead of benchmarking these methods and reporting scores over ranges of hyperparameters, we provide a discussion of the perceived gap in theory and practice for this family of uncertainty methods.

While the stochastic MCMC setting, estimating parameter updates from minibatches, is computationally convenient, it induces several theoretical challenges: i) minibatch noise introduced from small subsets of data [138], ii) omission of the Metropolis-Hastings correction step provides fundamentally biased estimates of posterior expectations [139], and

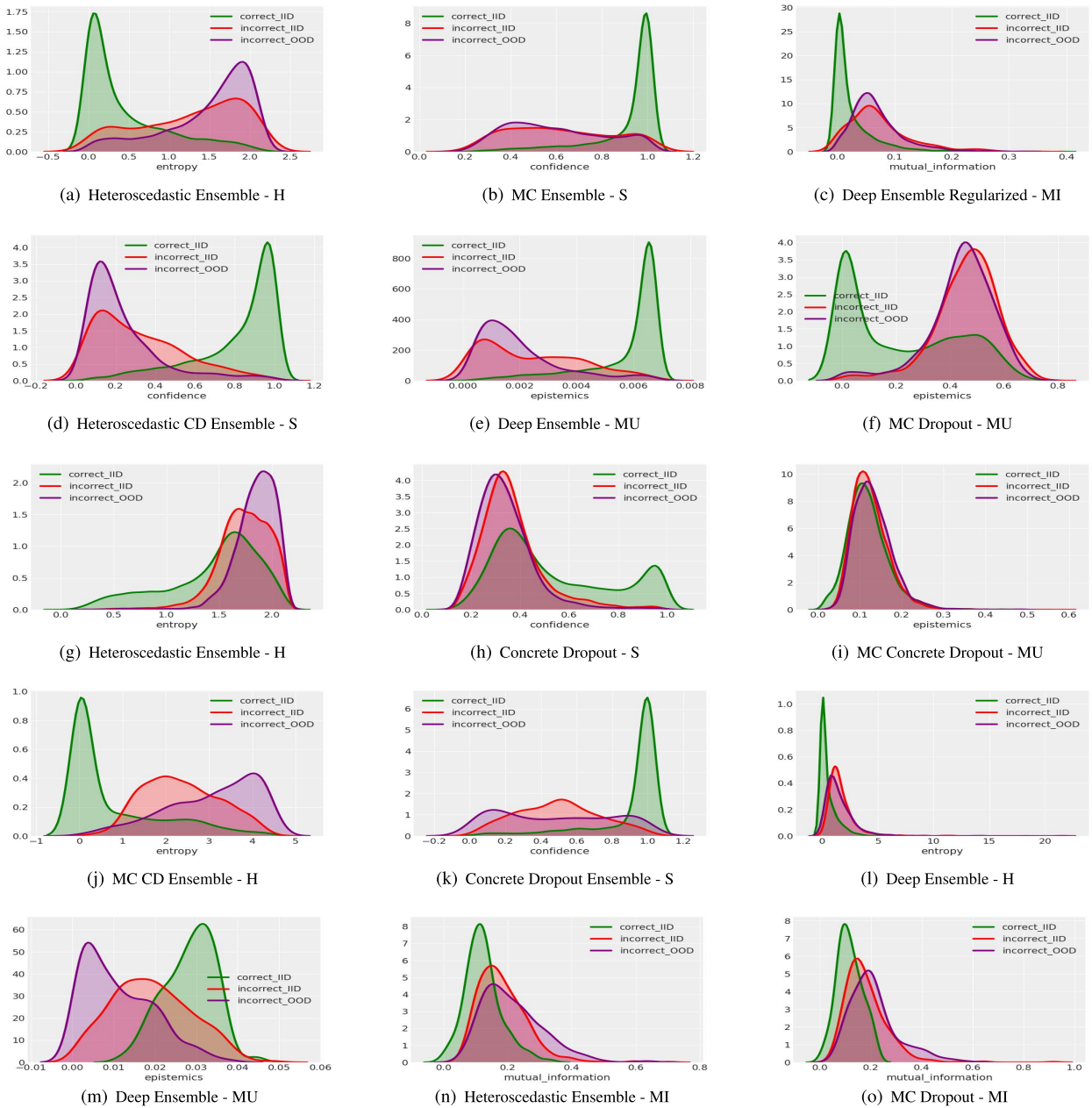


FIGURE 24. A selection of most interesting Gaussian kernel density plots over (abbreviated) model setup metrics evaluated on all datasets in row order 20news (a-c), CLINC150 (d-f), imdb (g-i), Reuters (j-l), AAPD (m-o). Each plot captures probabilistic density over correct ID (green), incorrect ID (red) and OOD (purple). From left to right, we have selected a high rank, middle rank, and low-rank method and uncertainty quantity combination. The density estimates demonstrate clear empirical difference over all datasets for various uncertainty quantities.

iii) the suggested practice of temperature tempering implies an approximation to the exact posterior instead of proper convergence [140], [141].

Closer to practice, [60]’s methods have been successfully benchmarked [15], [140] with reported performance on OOD detection for image classification datasets comparable to or better than Deep Ensembles. An important caveat is that all hyperparameters have been meticulously finetuned to the task at hand. This is non-trivial given the additional specification of the number of cycles as guided by a training budget,

proportion of burn-in steps, and finding an appropriately tempered posterior. The original work [60] mentions little dependence of results on these modifications to the optimization procedure, yet we observed similar to [141] “the complexity and fragility of hyper-parameter tuning, including the learning rate schedule and those that govern the simulation of a second-order Langevin dynamics”. Additionally, making combinations of uncertainty methods with cSG-MCMC is non-trivial, since regularization in any form influences the large scale curvature of the regions the optimizer explores.

TABLE 16. Comparison over in-domain setup for SNGP TextCNNv2 models. Results presented as mean \pm std. dev. across 5 runs for non-ensembles.

Method	Acc	NLL \downarrow	ECE \downarrow	Brier \downarrow	Softmax μ	Entropy μ	MU μ	DU μ
Unregularized	0.644 \pm 0.237	1.735 \pm 1.539	0.128 \pm 0.149	0.507 \pm 0.332	0.775 \pm 0.178	1.054 \pm 0.734	0	0
Regularized	0.640 \pm 0.249	1.843 \pm 1.677	0.102 \pm 0.107	0.531 \pm 0.363	0.711 \pm 0.202	1.685 \pm 1.350	0	0
MC Dropout	0.644 \pm 0.243	1.830 \pm 1.595	0.101 \pm 0.103	0.515 \pm 0.347	0.747 \pm 0.182	1.117 \pm 0.778	0.170 \pm 0.073	0
Concrete Dropout	0.568 \pm 0.339	1.820 \pm 1.595	0.070 \pm 0.056	0.612 \pm 0.487	0.661 \pm 0.238	1.039 \pm 0.660	0	0
MC Concrete Dropout	0.568 \pm 0.339	1.789 \pm 1.584	0.068 \pm 0.063	0.616 \pm 0.494	0.631 \pm 0.228	1.177 \pm 0.640	0.128 \pm 0.022	0
Deep Ensemble	0.665 \pm 0.260	1.614 \pm 1.643	0.110 \pm 0.145	0.470 \pm 0.347	0.755 \pm 0.201	1.177 \pm 0.873	0.020 \pm 0.012	0
Deep Ensemble Regularized	0.654 \pm 0.277	1.757 \pm 1.825	0.118 \pm 0.106	0.507 \pm 0.391	0.696 \pm 0.222	1.786 \pm 1.482	0.016 \pm 0.010	0
MC Ensemble	0.660 \pm 0.271	1.680 \pm 1.705	0.092 \pm 0.090	0.485 \pm 0.371	0.731 \pm 0.204	1.228 \pm 0.873	0.020 \pm 0.013	0
Concrete Dropout Ensemble	0.587 \pm 0.384	1.664 \pm 1.733	0.062 \pm 0.050	0.581 \pm 0.547	0.660 \pm 0.265	1.039 \pm 0.687	0.015 \pm 0.013	0
MC Concrete Dropout Ensemble	0.587 \pm 0.383	1.662 \pm 1.720	0.076 \pm 0.056	0.589 \pm 0.554	0.633 \pm 0.253	1.172 \pm 0.637	0.015 \pm 0.013	0
SNGP	0.644 \pm 0.237	1.748 \pm 1.572	0.131 \pm 0.148	0.512 \pm 0.342	0.773 \pm 0.183	1.061 \pm 0.757	0	0
Regularized SNGP	0.603 \pm 0.250	1.928 \pm 1.608	0.191 \pm 0.178	0.638 \pm 0.477	0.836 \pm 0.199	1.075 \pm 0.979	0	0
MC Dropout SNGP	0.624 \pm 0.242	1.840 \pm 1.612	0.120 \pm 0.111	0.541 \pm 0.349	0.695 \pm 0.189	1.432 \pm 0.870	0	0
Concrete Dropout SNGP	0.617 \pm 0.250	1.856 \pm 1.596	0.195 \pm 0.195	0.605 \pm 0.461	0.844 \pm 0.191	1.002 \pm 0.889	0	0
MC Concrete Dropout SNGP	0.634 \pm 0.240	1.792 \pm 1.595	0.117 \pm 0.122	0.526 \pm 0.346	0.718 \pm 0.185	1.315 \pm 0.771	0	0
Deep Ensemble SNGP	0.670 \pm 0.261	1.595 \pm 1.677	0.106 \pm 0.140	0.463 \pm 0.357	0.756 \pm 0.206	1.155 \pm 0.967	0.020 \pm 0.012	0
Deep Ensemble Regularized SNGP	0.630 \pm 0.282	1.716 \pm 1.758	0.187 \pm 0.257	0.572 \pm 0.509	0.833 \pm 0.216	1.247 \pm 1.231	0.027 \pm 0.017	0
MC Ensemble SNGP	0.652 \pm 0.273	1.686 \pm 1.710	0.141 \pm 0.133	0.494 \pm 0.368	0.691 \pm 0.208	1.489 \pm 0.927	0.018 \pm 0.011	0
Concrete Dropout Ensemble SNGP	0.645 \pm 0.278	1.665 \pm 1.739	0.162 \pm 0.232	0.529 \pm 0.457	0.816 \pm 0.214	1.187 \pm 1.096	0.025 \pm 0.016	0
MC Concrete Dropout Ensemble SNGP	0.658 \pm 0.267	1.673 \pm 1.708	0.125 \pm 0.104	0.493 \pm 0.371	0.697 \pm 0.204	1.438 \pm 0.878	0.018 \pm 0.011	0

only a single forward pass suffices without MC sampling to estimate the predictive distribution. Empirically, SNGP was shown to outperform Deep Ensemble by some margin on OOD detection for both image and text data. By demonstrating the relative importance of the decision boundary of a single model $f_{\theta}(y|x)$ versus averaging over multiple models, we are inspired to analyze the combination of SNGP with alternate uncertainty methods.

We have re-implemented SNGP using components of `edward2` [145], Laplace approximation, random feature GP and spectral normalization. In our experience, the most crucial hyperparameters to finetune were the number of inducing points ($\iota \leq 1024$) and spectral norm multiplier s . For the latter, we follow the recommended tuning procedure to find an appropriate value in the range $\{1, 2, 5, (10, 15)\}$, where we heuristically increased the search space.

For simplicity and computational reasons, we use TextCNN as base architecture. However, in order to correctly apply spectral normalization to convolutional filters [146], we had to re-implement TextCNN(v2) with 2D convolutions and maxpooling. This in turn requires specifying a fixed sequence length in advance, which invalidates directly comparing to the experiment results of Section IV. We additionally re-train base models with TextCNN(v2) and combine SNGP with our Regularized baseline (Reg), with MC Dropout (MCD), Concrete Dropout (CD) and Ensemble (Ens). For SNGP ensembles, we empirically selected $s = 15$ for the base model.

1) SNGP RESULTS

First, we present critical difference analyses for in-domain classification (Fig. 18) and novelty detection (Fig. 19). Ensembling SNGP models, *Deep Ensemble SNGP*, proves superior in-domain, followed by *Concrete Dropout Ensemble* with and without SNGP. For novelty detection, (MC) *Deep Ensemble* is most successful with small differences between next high-ranked methods.

To our surprise, *SNGP* ranks quite low on the text classification tasks, although in the original work it demonstrated OOD detection superior to *Deep Ensemble*. In what follows, we analyze the novelty detection ranking of SNGP, specifically per dataset and for multiple values of s .

In order to zoom in on the relative ranking of SNGP (combination) methods, we plot in Fig. 20 AUROC detection scores for datasets with interesting trend changes. Overall, SNGP underperforms on CLINC-OOS, with the exception of *Deep Ensemble SNGP*. For 20news, *SNGP* and *Deep Ensemble SNGP* rank high, although any additional regularization with *SNGP* worsens detection, even as ensemble. For Reuters, we observe the exact opposite to 20news, with *SNGP* reporting high detection scores only when regularization is added, e.g. *Regularized SNGP*. Remarkably, this trend is reversed for the base model, with *Unregularized* scoring particularly good.

Finally, Fig. 21 reports on how novelty detection varies for different values of the spectral normalization multiplier s . As the trend lines indicate, larger values of s generally improve novelty detection, although AUROC varies more (larger shading) between methods and datasets. This observation prompts us to investigate the optimality of s per dataset. The right subplot shows that spectral norm multipliers are very dataset-dependent and that searching further than the originally suggested range can give great performance boosts.

2) SNGP DISCUSSION

While *SNGP* was reported to outperform Deep Ensemble in the original CLINC OOD detection experiments [63], our results do not deliver the same ranking. While investigating the interaction of *SNGP* with different uncertainty methods, we observe the nontrivial role of spectral normalization, specifically setting the norm multiplier s to an appropriate value. Additionally, we contribute the analysis of the interplay with additional regularization mechanisms,

TABLE 17. Comparison over novelty detection setup for SNGP TextCNNv2 models. Results presented as mean ± std. dev. across 5 runs for non-ensembles.

Method	AUPR	AUROC	Softmax	DU	MU	Entropy	MI
Unregularized	0.625 ± 0.040	0.726 ± 0.118	0.433 ± 0.199	0	0	0.393 ± 0.224	0
Regularized	0.626 ± 0.036	0.724 ± 0.113	0.409 ± 0.219	0	0	0.445 ± 0.233	0
MC Dropout	0.625 ± 0.036	0.721 ± 0.112	0.398 ± 0.199	0	0.249 ± 0.152	0.406 ± 0.221	0.283 ± 0.187
Concrete Dropout	0.620 ± 0.042	0.716 ± 0.113	0.398 ± 0.202	0	0	0.449 ± 0.195	0
MC Concrete Dropout	0.621 ± 0.042	0.719 ± 0.113	0.405 ± 0.203	0	0.214 ± 0.141	0.464 ± 0.199	0.192 ± 0.182
Deep Ensemble	0.629 ± 0.041	0.737 ± 0.129	0.462 ± 0.227	0	0.480 ± 0.222	0.417 ± 0.253	0.334 ± 0.226
Deep Ensemble Regularized	0.627 ± 0.041	0.731 ± 0.127	0.428 ± 0.248	0	0.370 ± 0.315	0.464 ± 0.261	0.308 ± 0.266
MC Ensemble	0.628 ± 0.039	0.731 ± 0.125	0.428 ± 0.228	0	0.374 ± 0.296	0.428 ± 0.248	0.362 ± 0.225
Concrete Dropout Ensemble	0.617 ± 0.055	0.719 ± 0.134	0.412 ± 0.256	0	0.384 ± 0.314	0.471 ± 0.236	0.282 ± 0.258
MC Concrete Dropout Ensemble	0.619 ± 0.053	0.723 ± 0.133	0.415 ± 0.254	0	0.425 ± 0.271	0.480 ± 0.237	0.275 ± 0.241
SNGP	0.620 ± 0.043	0.717 ± 0.113	0.407 ± 0.196	0	0.123 ± 0.124	0.335 ± 0.255	0
Regularized SNGP	0.616 ± 0.046	0.705 ± 0.109	0.351 ± 0.169	0	0.206 ± 0.188	0.303 ± 0.191	0
MC Dropout SNGP	0.622 ± 0.039	0.715 ± 0.103	0.381 ± 0.169	0	0.195 ± 0.182	0.321 ± 0.206	0.245 ± 0.184
Concrete Dropout SNGP	0.613 ± 0.049	0.706 ± 0.116	0.357 ± 0.185	0	0.166 ± 0.182	0.313 ± 0.199	0
MC Concrete Dropout SNGP	0.617 ± 0.047	0.710 ± 0.114	0.382 ± 0.195	0	0.217 ± 0.203	0.329 ± 0.224	0.244 ± 0.189
Deep Ensemble SNGP	0.623 ± 0.046	0.729 ± 0.127	0.429 ± 0.231	0	0.467 ± 0.226	0.371 ± 0.301	0.338 ± 0.273
Deep Ensemble Regularized SNGP	0.620 ± 0.049	0.719 ± 0.120	0.398 ± 0.192	0	0.385 ± 0.191	0.348 ± 0.228	0.289 ± 0.235
MC Ensemble SNGP	0.627 ± 0.044	0.730 ± 0.117	0.415 ± 0.199	0	0.441 ± 0.186	0.348 ± 0.249	0.317 ± 0.251
Concrete Dropout Ensemble SNGP	0.620 ± 0.051	0.721 ± 0.128	0.414 ± 0.222	0	0.356 ± 0.260	0.351 ± 0.241	0.246 ± 0.293
MC Concrete Dropout Ensemble SNGP	0.625 ± 0.046	0.729 ± 0.122	0.430 ± 0.218	0	0.401 ± 0.251	0.359 ± 0.260	0.278 ± 0.286

which was missing in the literature. The original work mentions that given an approximation with the power iteration method, there is not a precise control of the true spectral norm. Whereas spectral normalization keeps the magnitude of updates to weights in check, Dropout regularization and weight decay may rescale layers’ spectral norm in unexpected ways. We hope our experimentation demonstrates the need for deeper understanding of how to combine multiple regularization mechanisms and maintain a good spectral norm approximation for effective posterior approximation.

**APPENDIX D
DETAILED EXPERIMENT RESULTS**

A. ZOOM-IN BENCHMARK EVIDENCE

In this Subsection we report additional evidence in support of our results, which did not suit the main manuscript.

B. ABSOLUTE BENCHMARK RESULTS

Next to reporting critical differences to analyze the relative performance of uncertainty methods, we also report results as summary statistics, following the methodology of [15]. Firstly, we report performance averaged over both runs and datasets, with the standard deviation over datasets. We indicate the best mean performance in bold. For various metrics the standard deviation is very large, which shows that the average over datasets for our benchmark would be a poor measure of central tendency. Since we benchmark on three multiclass and two multilabel datasets, any aggregate would be biased towards multiclass performance, hence why we specifically opted for rank and critical difference to analyze relative performance of each method.

Additionally, we compute the performance averaged over datasets, with the standard deviation over multiple runs for all individual models. All raw model results are available at https://github.com/Jordy-VL/uncertainty-bench/tree/main/experiments/raw_results.

1) AVERAGED OVER DATASETS AND RUNS

See Tables 7–12.

2) AVERAGED OVER RUNS

See Tables 13–17.

REFERENCES

- [1] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, M. Shahzad, W. Yang, R. Bamler, and X. Xiang Zhu, “A survey of uncertainty in deep neural networks,” 2021, *arXiv:2107.03342*.
- [2] A. Y. Foong, Y. Li, J. M. Hernández-Lobato, and R. E. Turner, “In-between uncertainty in Bayesian neural networks,” in *Proc. ICML Workshop Uncertainty Robustness Deep Learn.*, 2019, pp. 1–31.
- [3] M. A. F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, “A review of novelty detection,” *Signal Process.*, vol. 99, pp. 215–249, Jun. 2014.
- [4] L. Valentin Jospin, W. Buntine, F. Boussaid, H. Laga, and M. Bennamoun, “Hands-on Bayesian neural networks—A tutorial for deep learning users,” 2020, *arXiv:2007.06823*.
- [5] Z. C. Lipton and J. Steinhardt, “Troubling trends in machine learning scholarship,” 2018, *arXiv:1807.03341*.
- [6] N. Seadat and C. Kanan, “Towards calibrated and scalable uncertainty representations for neural networks,” 2019, *arXiv:1911.00104*.
- [7] A. G. Wilson, “The case for Bayesian deep learning,” 2020, *arXiv:2001.10995*.
- [8] J. Mukhoti, P. Stenatorp, and Y. Gal, “On the importance of strong baselines in Bayesian deep learning,” 2018, *arXiv:1811.09385*.
- [9] J. M. Hernández-Lobato and R. Adams, “Probabilistic backpropagation for scalable learning of Bayesian neural networks,” in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1861–1869.
- [10] T. Pearce, F. Leibfried, and A. Brintrup, “Uncertainty in neural networks: Approximately Bayesian ensembling,” in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 234–244.
- [11] A. Filos, S. Farquhar, A. N. Gomez, T. G. J. Rudner, Z. Kenton, L. Smith, M. Alizadeh, A. de Kroon, and Y. Gal, “A systematic comparison of Bayesian deep learning robustness in diabetic retinopathy tasks,” 2019, *arXiv:1912.10481*.
- [12] Y. Ovadia, S. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek, “Can you trust your model’s uncertainty? Evaluating predictive uncertainty under dataset shift,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 13991–14002.
- [13] W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson, “A simple baseline for Bayesian uncertainty in deep learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 13153–13164.

- [14] A. Ashukha, A. Lyzhov, D. Molchanov, and D. Vetrov, "Pitfalls of in-domain uncertainty estimation and ensembling in deep learning," in *Proc. Int. Conf. Learn. Represent.*, Jul. 2019. [Online]. Available: <https://dblp.org/rec/conf/iclr/AshukhaLMV20.html?view=bibtex>
- [15] M. P. Vadera, A. D. Cobb, B. Jalaian, and B. M. Marlin, "URSABench: Comprehensive benchmarking of approximate Bayesian inference methods for deep neural networks," 2020, *arXiv:2007.04466*.
- [16] J. Gilmer, R. P. Adams, I. Goodfellow, D. Andersen, and G. E. Dahl, "Motivating the rules of the game for adversarial example research," 2018, *arXiv:1807.06732*.
- [17] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *Proc. 45th Annu. Meeting Assoc. Comput. Linguistics*, 2007, pp. 440–447.
- [18] H. Daumé III, "Frustratingly easy domain adaptation," in *Proc. 45th Annu. Meeting Assoc. Comput. Linguistics*, Prague, Czech Republic, Jun. 2007, pp. 1–8.
- [19] M. Joshi, M. Dredze, W. Cohen, and C. Rose, "Multi-domain learning: When do domains matter?" in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn.*, 2012, pp. 1302–1312.
- [20] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, 2017.
- [21] Y. Ziser and R. Reichart, "Neural structural correspondence learning for domain adaptation," in *Proc. 21st Conf. Comput. Natural Lang. Learn. (CoNLL)*, 2017, pp. 1–11.
- [22] A. Ramponi and B. Plank, "Neural unsupervised domain adaptation in NLP—A survey," in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 6838–6855.
- [23] M. Arjovsky, "Out of distribution generalization in machine learning," Ph.D. dissertation, Courant Inst. Math. Sci., New York Univ., New York, NY, USA, 2020.
- [24] J. Wen, N. Zheng, J. Yuan, Z. Gong, and C. Chen, "Bayesian uncertainty matching for unsupervised domain adaptation," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3849–3855.
- [25] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–27.
- [26] W. Liu, X. Wang, J. Owens, and Y. Li, "Energy-based out-of-distribution detection," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 21464–21475.
- [27] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan, "Do deep generative models know what they don't know?" in *Proc. Int. Conf. Learn. Represent.*, 2018. [Online]. Available: <https://openreview.net/forum?id=H1xwNhCcYm>
- [28] Y. Geifman and R. El-Yaniv, "Selective classification for deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Red Hook, NY, USA: Curran Associates, 2017, pp. 1–10.
- [29] S. Rabanser, S. Günnemann, and Z. Lipton, "Failing loudly: An empirical study of methods for detecting dataset shift," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2019, pp. 1396–1408.
- [30] A. Shafaei, M. Schmidt, and J. Little, "A less biased evaluation of out-of-distribution sample detectors," in *Proc. BMVC*, 2019. [Online]. Available: https://explore.openaire.eu/search/publication?articleId=arXiv_:::16ed651f32a1fccab8616ec2cbb517d2
- [31] S. Bulusu, B. Kailkhura, B. Li, P. K. Varshney, and D. Song, "Anomalous example detection in deep learning: A survey," *IEEE Access*, vol. 8, pp. 132330–132347, 2020.
- [32] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Inf. Fusion*, vol. 76, pp. 243–297, Dec. 2021.
- [33] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian SegNet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," 2015, *arXiv:1511.02680*.
- [34] L. Mou, H. Zhou, and L. Li, "Discreteness in neural natural language processing," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP-IJCNLP)*, Hong Kong, Nov. 2019, pp. 1–72.
- [35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
- [36] M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling modern machine-learning practice and the classical bias–variance trade-off," *Proc. Nat. Acad. Sci. USA*, vol. 116, no. 32, pp. 15849–15854, 2019.
- [37] D. Roth, "Learning to resolve natural language ambiguities: A unified approach," in *Proc. AAAI/IAAI*, 1998, pp. 806–813.
- [38] L. Wan, G. Papageorgiou, M. Seddon, and M. Bernardoni, "Long-length legal document classification," 2019, *arXiv:1912.06905*.
- [39] W. Chen, Y. Su, Y. Shen, Z. Chen, X. Yan, and W. Y. Wang, "How large a vocabulary does text classification need? A variational approach to vocabulary selection," in *Proc. Conf. North*, 2019, pp. 1–11.
- [40] A. K. McCallum, "Multi-label text classification with a mixture model trained by EM," in *Proc. AAAI Workshop Text Learn.*, 1999.
- [41] Y. Xiao and W. Y. Wang, "Quantifying uncertainties in natural language processing tasks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 7322–7329.
- [42] D. Hendrycks, X. Liu, E. Wallace, A. Dziedzic, R. Krishnan, and D. Song, "Pretrained transformers improve out-of-distribution robustness," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 1–11.
- [43] X. Zhang, F. Chen, C.-T. Lu, and N. Ramakrishnan, "Mitigating uncertainty in document classification," in *Proc. Conf. North*, Jun. 2019, pp. 1–11.
- [44] S. Mukherjee and A. H. Awadallah, "Uncertainty-aware self-training for few-shot text classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 21199–21212.
- [45] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul./Oct. 1948.
- [46] H. Zaragoza and F. d'Alché-Buc, "Confidence measures for neural network classifiers," in *Proc. 7th Int. Conf. Process. Manage. Uncertainty Knowl. Based Syst.*, 1998, pp. 886–893.
- [47] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1321–1330.
- [48] A. Shrikumar and A. Kundaje, "Maximum likelihood with bias-corrected calibration is hard-to-beat at label shift adaptation," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 222–232.
- [49] J. Denker, D. Schwartz, B. Wittner, S. Solla, R. Howard, L. Jackel, and J. Hopfield, "Large automatic learning, rule extraction, and generalization," *Complex Syst.*, vol. 1, no. 5, pp. 877–922, 1987.
- [50] D. J. MacKay, "Bayesian methods for adaptive models," Ph.D. dissertation, Dept. Comput. Math. Sci., California Inst. Technol., Pasadena, CA, USA, 1992.
- [51] R. M. Neal, "Bayesian mixture modeling," in *Maximum Entropy Bayesian Methods*. Cham, Switzerland: Springer, 1992, pp. 197–211.
- [52] G. E. Hinton and D. van Camp, "Keeping the neural networks simple by minimizing the description length of the weights," in *Proc. 6th Annu. Conf. Comput. Learn. Theory (COLT)*, 1993, pp. 5–13.
- [53] D. J. C. Mackay, "Probable networks and plausible predictions—A review of practical Bayesian methods for supervised neural networks," *Netw., Comput. Neural Syst.*, vol. 6, no. 3, pp. 469–505, Jan. 1995.
- [54] Z. Ghahramani, "A history of Bayesian neural networks," in *Proc. NIPS Workshop Bayesian Deep Learn.*, 2016.
- [55] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 6402–6413.
- [56] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, "Snapshot ensembles: Train 1, get M for free," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–14.
- [57] T. Garipov, P. Izmailov, D. Podoprikin, D. P. Vetrov, and A. G. Wilson, "Loss surfaces, mode connectivity, and fast ensembling of DNNs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8789–8798.
- [58] W. Gilks, S. Richardson, and D. Spiegelhalter, *Markov Chain Monte Carlo in Practice* (Chapman & Hall/CRC Interdisciplinary Statistics). Oxfordshire, U.K.: Taylor & Francis, 1995.
- [59] M. Hoffman, "Langevin dynamics as nonparametric variational inference," in *Proc. 2nd Symp. Adv. Approx. Bayesian Inference*, 2019.
- [60] R. Zhang, C. Li, J. Zhang, C. Chen, and A. G. Wilson, "Cyclical stochastic gradient MCMC for Bayesian deep learning," in *Proc. Int. Conf. Learn. Represent.*, 2020.

- [61] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.
- [62] J. Van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal, "Uncertainty estimation using a single deep deterministic neural network," in *Proc. 37th Int. Conf. Mach. Learn.*, H. D. III and A. Singh, Eds. Jul. 2020, pp. 9690–9700.
- [63] J. Z. 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 *Proc. Neural Inf. Process. Syst.*, 2020, pp. 7498–7512.
- [64] R. E. Turner and M. Sahani, "Two problems with variational expectation maximisation for time-series models," in *Bayesian Time Series Models*, D. Barber, T. Cemgil, and S. Chiappa, Eds. Cambridge, U.K.: Cambridge Univ. Press, 2011, ch. 5, pp. 109–130.
- [65] Y. Wen, D. Tran, and J. Ba, "Batchensemble: An alternative approach to efficient ensemble and lifelong learning," in *Proc. Int. Conf. Learn. Represent.*, 2019.
- [66] F. K. Gustafsson, M. Danelljan, and T. B. Schon, "Evaluating scalable Bayesian deep learning methods for robust computer vision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 318–319.
- [67] A. Krogh and J. Vedelsby, "Neural network ensembles, cross validation, and active learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 7, G. Tesauro, D. Touretzky, and T. Leen, Eds. Cambridge, MA, USA: MIT Press, 1995, pp. 231–238.
- [68] S. Jain, G. Liu, J. Mueller, and D. Gifford, "Maximizing overall diversity for improved uncertainty estimates in deep ensembles," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 4264–4271.
- [69] B. Brazowski and E. Schneidman, "Collective learning by ensembles of altruistic diversifying neural networks," 2020, *arXiv:2006.11671*.
- [70] Y. Gal, J. Hron, and A. Kendall, "Concrete dropout," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3581–3590.
- [71] I. Osband, "Risk versus uncertainty in deep learning: Bayes, bootstrap and the dangers of dropout," in *Proc. NIPS Workshop Bayesian Deep Learn.*, vol. 192, 2016.
- [72] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5574–5584.
- [73] Y. Kwon, J.-H. Won, B. J. Kim, and M. C. Paik, "Uncertainty quantification using Bayesian neural networks in classification: Application to biomedical image segmentation," *Comput. Statist. Data Anal.*, vol. 142, Feb. 2020, Art. no. 106816.
- [74] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," 2019, *arXiv:1910.09457*.
- [75] S. Depeweg, J.-M. Hernandez-Lobato, F. Doshi-Velez, and S. Udluft, "Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1184–1193.
- [76] L. Smith and Y. Gal, "Understanding measures of uncertainty for adversarial example detection," in *Proc. Conf. Uncertainty Artif. Intell. (UAI)*, 2018, pp. 1–10.
- [77] A. Malinin, B. Mlodozienic, and M. Gales, "Ensemble distribution distillation," in *Proc. Int. Conf. Learn. Represent.*, 2019.
- [78] M. Vadera, B. Jalaian, and B. Marlin, "Generalized Bayesian posterior expectation distillation for deep neural networks," in *Proc. Conf. Uncertainty Artif. Intell.*, 2020, pp. 719–728.
- [79] S. Fort, H. Hu, and B. Lakshminarayanan, "Deep ensembles: A loss landscape perspective," 2019, *arXiv:1912.02757*.
- [80] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," in *Proc. Adv. neural Inf. Process. Syst.*, vol. 31, 2018, pp. 6391–6401.
- [81] S. Fort and S. Jastrzebski, "Large scale structure of neural network loss landscapes," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Álché-Buc, E. Fox, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2019, pp. 6709–6717.
- [82] R. Neal, *Bayesian Learning for Neural Networks*. New York, NY, USA: Springer, 1996.
- [83] E. Nalisnick, J. M. Hernández-Lobato, and P. Smyth, "Dropout as a structured shrinkage prior," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4712–4722.
- [84] A. Gotmare, N. S. Keskar, C. Xiong, and R. Socher, "Using mode connectivity for loss landscape analysis," in *Proc. ICLR Workshop Modern Trends Nonconvex Optim. Mach. Learn.*, 2018.
- [85] V. Fortuin, "Priors in Bayesian deep learning: A review," 2021, *arXiv:2105.06868*.
- [86] M. H. DeGroot and S. E. Fienberg, "The comparison and evaluation of forecasters," *J. Roy. Stat. Soc., Ser. D Statistician*, vol. 32, nos. 1–2, pp. 12–22, 1983.
- [87] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2002, pp. 694–699.
- [88] M. P. Naeni, G. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using Bayesian binning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 29, no. 1, 2015, pp. 1–7.
- [89] A. Kumar, P. Liang, and T. Ma, "Verified uncertainty calibration," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 3792–3803.
- [90] M. Kull, M. P. Nieto, M. Kängsepp, T. S. Filho, H. Song, and P. Flach, "Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with Dirichlet calibration," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 12316–12326.
- [91] J. Wenger, H. Kjellström, and R. Triebel, "Non-parametric calibration for classification," in *Proc. 23rd Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2020, pp. 178–190.
- [92] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *J. Statist. Planning Inference*, vol. 90, no. 2, pp. 227–244, 2000.
- [93] S. Bickel, M. Brückner, and T. Scheffer, "Discriminative learning under covariate shift," *J. Mach. Learn. Res.*, vol. 10, no. 9, pp. 1–19, 2009.
- [94] S. Park, O. Bastani, J. Weimer, and I. Lee, "Calibrated prediction with covariate shift via unsupervised domain adaptation," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 3219–3229.
- [95] J. Duchi and H. Namkoong, "Learning models with uniform performance via distributionally robust optimization," 2018, *arXiv:1810.08750*.
- [96] H. Namkoong, *Reliable Machine Learning Via Distributional Robustness*. Stanford, CA, USA: Stanford Univ., 2019.
- [97] A. Pampari and S. Ermon, "Unsupervised calibration under covariate shift," 2020, *arXiv:2006.16405*.
- [98] X. Wang, M. Long, J. Wang, and M. Jordan, "Transferable calibration with lower bias and variance in domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds. Red Hook, NY, USA: Curran Associates, 2020, pp. 19212–19223.
- [99] Y. Gong, X. Lin, Y. Yao, T. G. Dietterich, A. Divakaran, and M. Gervasio, "Confidence calibration for domain generalization under covariate shift," 2021, *arXiv:2104.00742*.
- [100] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera, "A unifying view on dataset shift in classification," *Pattern Recognit.*, vol. 45, no. 1, pp. 521–530, 2012.
- [101] M. Markou and S. Singh, "Novelty detection: A review—Part 1: Statistical approaches," *Signal Process.*, vol. 83, no. 12, pp. 2481–2497, 2003.
- [102] T. Poggio, V. Torre, and C. Koch, "Computational vision and regularization theory," *Readings Comput. Vis.*, pp. 638–643, Jan. 1987.
- [103] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities Complementarity Problems*. Cham, Switzerland: Springer, 2007.
- [104] A. K. Tanwani and M. Farooq, "Classification potential vs. classification accuracy: A comprehensive study of evolutionary algorithms with biomedical datasets," in *Learning Classifier Systems*. Cham, Switzerland: Springer, 2009, pp. 127–144.
- [105] K. Lang, "Newsweeder: Learning to filter netnews. version 20news-18828," in *Machine Learning Proceedings*. Burlington, MA, USA: Morgan Kaufmann, 1995, pp. 331–339.
- [106] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *Proc. 5th Int. Conf. Learn. Represent.*, 2017.
- [107] Q. Diao, M. Qiu, C.-Y. Wu, A. J. Smola, J. Jiang, and C. Wang, "Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS)," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 193–202.

- [108] S. Larson, A. Mahendran, J. J. Peper, C. Clarke, A. Lee, P. Hill, J. K. Kummerfeld, K. Leach, M. A. Laurenzano, L. Tang, and J. Mars, "An evaluation dataset for intent classification and out-of-scope prediction," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 1311–1316.
- [109] C. Apté, F. Damerou, and S. M. Weiss, "Automated learning of decision rules for text categorization," *ACM Trans. Inf. Syst.*, vol. 12, no. 3, pp. 233–251, Jul. 1994.
- [110] P. Yang, X. Sun, W. Li, S. Ma, W. Wu, and H. Wang, "SGM: Sequence generation model for multi-label classification," in *Proc. 27th Int. Conf. Comput. Linguistics*, Santa Fe, NM, USA, Aug. 2018, pp. 1–12.
- [111] A. Adhikari, A. Ram, R. Tang, W. L. Hamilton, and J. Lin, "Exploring the limits of simple learners in knowledge distillation for document classification with DocBERT," in *Proc. 5th Workshop Represent. Learn. (NLP)*, 2020, pp. 72–77.
- [112] C. Du, H. Sun, J. Wang, Q. Qi, and J. Liao, "Adversarial and domain-aware BERT for cross-domain sentiment analysis," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 4019–4028.
- [113] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1746–1751.
- [114] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," 2015, *arXiv:1510.03820*.
- [115] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North American Chapter Assoc. Comput. Linguistics*, Jun. 2019, pp. 1–16.
- [116] J. Nixon, M. W. Dusenberry, L. Zhang, G. Jerfel, and D. Tran, "Measuring calibration in deep learning," in *Proc. CVPR Workshops*, vol. 2, no. 7, Jun. 2019, pp. 1–4.
- [117] D. Widmann, F. Lindsten, and D. Zachariah, "Calibration tests in multi-class classification: A unifying framework," in *Proc. 32th Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 12236–12246.
- [118] D. Widmann, F. Lindsten, and D. Zachariah, "Calibration tests beyond classification," in *Proc. Int. Conf. Learn. Represent.*, 2021.
- [119] A. H. Murphy and R. L. Winkler, "Scoring rules in probability assessment and evaluation," *Acta Psycholog.*, vol. 34, pp. 273–286, Jan. 1970.
- [120] J. Hernandez-Orallo, P. A. Flach, and C. Ferri, "A unified view of performance metrics: Translating threshold choice into expected classification loss," *J. Mach. Learn. Res.*, vol. 13, pp. 2813–2869, Oct. 2012.
- [121] J. Quinero-Candela, C. E. Rasmussen, F. Sinz, O. Bousquet, and B. Schölkopf, "Evaluating predictive uncertainty challenge," in *Proc. Mach. Learn. Challenges Workshop*. Cham, Switzerland: Springer, 2005, pp. 1–27.
- [122] G. W. Brier, "Verification of forecasts expressed in terms of probability," *Monthly Weather Rev.*, vol. 78, no. 1, pp. 1–3, 1950.
- [123] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.
- [124] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2009.
- [125] F. Wu and Y. Huang, "Sentiment domain adaptation with multiple sources," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2016, pp. 301–310.
- [126] A. Emmott, S. Das, T. Dietterich, A. Fern, and W.-K. Wong, "A meta-analysis of the anomaly detection problem," 2015, *arXiv:1503.01158*.
- [127] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.
- [128] D. Amodei, C. Olah, J. Steinhardt, P. F. Christiano, J. Schulman, and D. Mané, "Concrete problems in AI safety," 2016, *arXiv:1606.06565*.
- [129] C. Leibig, V. Allken, M. S. Ayhan, P. Berens, and S. Wahl, "Leveraging uncertainty information from deep neural networks for disease detection," *Sci. Rep.*, vol. 7, no. 1, pp. 1–14, Dec. 2017.
- [130] S. Gururangan, S. Swayamdipta, O. Levy, R. Schwartz, S. Bowman, and N. A. Smith, "Annotation artifacts in natural language inference data," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., (Short Papers)*, vol. 2, 2018, pp. 1–6.
- [131] G. Scalia, C. A. Grambow, B. Pernici, Y.-P. Li, and W. H. Green, "Evaluating scalable uncertainty estimation methods for deep learning-based molecular property prediction," *J. Chem. Inf. Model.*, vol. 60, no. 6, pp. 2697–2717, Jun. 2020.
- [132] J. Caldeira and B. Nord, "Deeply uncertain: Comparing methods of uncertainty quantification in deep learning algorithms," *Mach. Learning, Sci. Technol.*, vol. 2, no. 1, Dec. 2020, Art. no. 015002.
- [133] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [134] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 4, J. E. Moody, S. J. Hanson, and R. P. Lippmann, Eds. Burlington, MA, USA: Morgan-Kaufmann, 1992, pp. 950–957.
- [135] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent.*, 2019.
- [136] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune BERT for text classification?" in *Proc. China Nat. Conf. Chin. Comput. Linguistics*. Cham, Switzerland: Springer, 2019, pp. 194–206.
- [137] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *Proc. Int. Conf. Learn. Represent.*, 2017.
- [138] R. Luo, J. Wang, Y. Yang, J. WANG, and Z. Zhu, "Thermostat-assisted continuously-tempered Hamiltonian Monte Carlo for Bayesian learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2018, pp. 10696–10705.
- [139] P. Izmailov, S. Vikram, M. D. Hoffman, and A. Gordon Wilson, "What are Bayesian neural network posteriors really like?" 2021, *arXiv:2104.14421*.
- [140] F. Wenzel, K. Roth, B. Veeling, J. Swiatkowski, L. Tran, S. Mandt, J. Snoek, T. Salimans, R. Jenatton, and S. Nowozin, "How good is the Bayes posterior in deep neural networks really?" in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 10248–10259.
- [141] G. Franzese, R. Candela, D. Milios, M. Filippone, and P. Michiardi, "Isotropic SGD: A practical approach to Bayesian posterior sampling," 2020, *arXiv:2006.05087*.
- [142] B.-H. Tran, S. Rossi, D. Milios, and M. Filippone, "All you need is a good functional prior for Bayesian deep learning," 2020, *arXiv:2011.12829*.
- [143] D. Duvenaud, O. Rippel, R. Adams, and Z. Ghahramani, "Avoiding pathologies in very deep networks," in *Proc. Artif. Intell. Statist.*, 2014, pp. 202–210.
- [144] Z. Lu, E. Ie, and F. Sha, "Uncertainty estimation with infinitesimal jackknife, its distribution and mean-field approximation," 2020, *arXiv:2006.07584*.
- [145] D. Tran, M. W. Dusenberry, D. Hafner, and M. van der Wilk, "Bayesian Layers: A module for neural network uncertainty," in *Proc. Neural Inf. Process. Syst.*, 2019, pp. 14660–14672.
- [146] H. Gouk, E. Frank, B. Pfahringer, and M. J. Cree, "Regularisation of neural networks by enforcing Lipschitz continuity," *Mach. Learn.*, vol. 110, no. 2, pp. 393–416, Feb. 2021.



JORDY VAN LANDEGHEM received the M.A. degree in linguistics and the M.Sc. degree in artificial intelligence from KU Leuven, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree in computer science. He completed research internships at Oracle and Nuance Communications. He is the Lead AI Researcher at Contract.fit. His industrial Ph.D. project titled "Intelligent Automation for AI-Driven Document Understanding" focuses on the fundamentals of probabilistic deep learning, with an emphasis on calibration, uncertainty quantification, and out-of-distribution robustness, in order to obtain more reliable machine learning systems.



MATTHEW BLASCHKO received the B.S. degree from Columbia University, the M.Sc. degree from the University of Massachusetts Amherst, the Ph.D. degree (*summa cum laude*) in electrical engineering and computer science from Technische Universität Berlin for work done at the Max Planck Institute for Biological Cybernetics, Tübingen, Germany, and the Habilitation degree from the École Normale Supérieure de Cachan, France. Subsequently, he was a Newton International Fel-

low at the Department of Engineering Science, University of Oxford. Since 2015, he has been a Professor with the Department of Electrical Engineering, KU Leuven, Belgium. Prior to joining KU Leuven, he was a Permanent Research Scientist at the INRIA Saclay Research Center and a Faculty Member at École Centrale Paris. His research interest includes machine learning techniques applied to visual data. He has received the Main Prize of the German Association for Pattern Recognition and the Best Paper Awards at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) and the European Conference on Computer Vision (ECCV).



MARIE-FRANCINE MOENS received the M.Sc. and Ph.D. degrees in computer science from KU Leuven. She is currently a Full Professor at the Department of Computer Science, KU Leuven. She is the Director of the Language Intelligence and Information Retrieval (LIIR) Research Laboratory. She holds the ERC Advanced Grant CALCULUS (2018–2024) granted by the European Research Council. Her main research interests include natural language processing, text and multimedia mining, machine learning, and information retrieval.

• • •



BERTRAND ANCKAERT received the M.Sc. and Ph.D. degrees in computer science from the University of Ghent. He has worked as a Researcher at Microsoft, in 2008, obtained two U.S. patents, and stayed on at the Boston Consulting Group, from 2008 to 2016, holding the title of a Principal. In 2016, he co-founded Contract.fit, a European software-as-a-service company that offers an intelligent automation solution using the latest AI/ML techniques. His research interests include scalable

ML systems for document understanding, with specific applications of key information extraction and email routing.