# Enhancing Transformer Architectures with Real-Time Uncertainty for Reliable Text Classification

Master's thesis submitted

to

**Prof. Dr. Stefan Lessmann**
**Prof. Dr. Benjamin Fabian**

Humboldt-Universität zu Berlin
School of Business and Economics
Chair of Information Systems

by

**Johann Benedikt Sonnenburg**
ID 597448

in partial fulfillment of the requirements
for the degree of
**Master of Science (M.Sc.) in Information Systems**

Berlin, June 10, 2024

## ABSTRACT

Despite their improved predictive power, modern deep learning architectures frequently suffer from miscalibration, a critical issue in real-world settings. This can be mitigated by enabling a model to accurately assess the uncertainty of its predictions, which is traditionally resource-intensive. This work addresses these closely connected problems in the context of the natural language processing domain, specifically for a hate speech detection task. For this purpose, we adapt the uncertainty-aware distribution distillation framework, which enables distilling a model that generates high-quality uncertainty estimates in real-time, to a transformer-based deep learning architecture. We empirically evaluate the predictive performance and the quality of the uncertainty estimates of the resulting teacher and student models and assess their robustness against covariate shifts and out-of-distribution data. Our findings reveal that we are able to successfully distill a student model that retains accuracy but also enhances calibration and uncertainty estimation quality, thereby significantly improving robustness. Thus, our work highlights the effective application of this distillation framework to transformer architectures and underscores its practical applicability to the natural language processing domain.

## Contents

## List of Figures

# 1 Introduction

Many real-world machine learning applications benefit from models that are not only accurate but also capable of indicating their confidence levels (Guo et al., 2017). Despite the advancements in the size and predictive power of modern deep learning architectures, these models have increasingly shown miscalibration issues. Miscalibration implies that the predicted probabilities of outcomes do not reflect their true likelihood, which can lead to overconfidence in incorrect predictions.

High-quality uncertainty estimates address this issue by providing a measure of confidence in the model's predictions (Guo et al., 2017). While the existing literature predominantly explores uncertainty quantification within the computer vision (CV) domain, particularly in safety-critical applications such as autonomous driving and medical imaging, many studies aim to apply the associated methodologies to other fields, including natural language understanding (NLU) (He et al., 2020; Hu et al., 2021; Van Landeghem et al., 2022). However, this area of research still remains underexplored (Van Landeghem et al., 2022).

The need for reliable real-time uncertainty estimation is a particularly demanding challenge. In many applications, having near-immediate access to the model's confidence levels is crucial to making timely and informed decisions. Traditional approaches for uncertainty estimation often involve computationally intensive sampling techniques, which are not feasible for real-time applications. Consequently, there is a need to develop novel methods that can provide accurate uncertainty estimates with minimal computational overhead.

Uncertainty-aware distribution distillation, proposed by Shen et al. (2021), is one such method. It builds on the foundation of knowledge distillation (Hinton et al., 2015) to generate a student model capable of providing high-quality uncertainty estimates with minimal latency, aiming to match those of a sampling-based teacher model.

Text classification is one NLU task where real-time uncertainty estimation is particularly relevant, especially within the social media domain (Van Landeghem et al., 2022). Detecting offensive or toxic language is a typical task for automated content moderation at scale. For this purpose, social media platforms often use a combination of machine learning models and human experts (Pruksachatkun et al., 2021; Gillespie, 2020). To prevent wrongful censorship or the oversight of offensive content, accurately identifying when a model is uncertain about its predictions is crucial (Fawcett, 2006). High-uncertainty cases can be escalated to human moderators, ensuring that the moderation process adheres to ethical standards and regulatory requirements (Binns, 2018; Barral Martínez, 2023). Real-time uncertainty quantification can thus enhance the reliability of automated content-moderation systems.

This thesis contributes to the existing literature by applying a novel real-time uncertainty quantification technique to the natural language processing (NLP) domain. Specifically, we implement the uncertainty-aware distribution distillation framework in a transformer-based NLU architecture for text classification. This work aims to thoroughly investigate the validity of this approach in the given context and demonstrate its potential for transformer-based architectures in the NLP domain.

Concretely, we evaluate the framework when applied to BERT (Bidirectional Encoder Representations from Transformers, Devlin et al. (2019)) for a hate speech detection task. In this context, we conduct an in-depth analysis of predictive performance and uncertainty quality, examine the robustness to covariate shifts, and perform an out-of-distribution (OOD) detection analysis. Together, these evaluations will provide a comprehensive understanding of the approach's effectiveness and practicality, allowing us to estimate its potential in the given setting.

The aims of this work are reflected by our research questions, which will guide all subsequent analyses:

1. Can we successfully apply uncertainty-aware distribution distillation, as proposed by Shen et al. (2021), to the BERT architecture?
   (a) In particular, can we train a student model based also on BERT that learns to predict the approximate posterior predictive distribution of the teacher model, which employs MC dropout to perform approximate Bayesian inference?
   (b) How does the student model's performance compare to its teacher when evaluated on the test dataset?
   (c) How does the uncertainty quality compare between teacher and student model?
2. How do the resulting teacher and student network fare with respect to predictive performance and uncertainty quality when subjected to different types and levels of distributional shift?
3. Extending the previous question, is there a difference in predictive uncertainty across the models when evaluated on OOD data?

The proposed topic is of great relevance to researchers and practitioners alike. Much contemporary machine learning research investigates deep learning methods, especially those applied to the NLU domain. Real-time uncertainty

quantification is a highly active area of research and has yet to be widely applied to NLU tasks. In particular, its use in combination with transformer architectures has yet to be explored in detail. Enhancing transformer-based models with real-time uncertainty capabilities promises increased robustness, interpretability, and reliability. Specifically, incorporating these capabilities into transformer-based models such as BERT can provide practitioners with tools to better manage predictive confidence and detect OOD inputs (Kendall and Gal, 2017). These insights are crucial, especially for systems employed by non-experts in real-world settings.

The remaining part of this thesis is structured as follows: Section 2 provides an overview of the related literature. In Sections 3 and 4, we describe fundamental theoretical concepts and the general experimental setup. Section 5 focuses on the application of the uncertainty-aware distribution distillation framework. Subsequently, Section 6 discusses the robustness study and additionally examines the impact of a student augmentation technique. Section 7 details the OOD analysis. Finally, Section 8 provides an in-depth discussion of our findings before Section 9 concludes this work.

## 2 Related Literature

The Bayesian paradigm naturally incorporates uncertainty over beliefs and has been successfully applied to machine learning methods. In Bayesian learning, we update our beliefs about model parameters using observed data by placing a prior distribution over the parameters, reflecting our initial beliefs. We define a likelihood function that describes how the data is generated from these parameters. Combining the prior and the likelihood through Bayes' theorem, we obtain the posterior distribution, which refines our parameter estimates based on the data (Gal, 2016).

Bayesian neural networks offer a probabilistic interpretation of deep learning models by placing priors over the weights and biases, thus allowing for uncertainty estimation of the outputs (MacKay, 1992). The computational intractability of posteriors in deep learning models requires techniques for approximate Bayesian inference.

Several techniques exist for approximating the posterior distribution over the model parameters, including several based on Markov chain Monte Carlo (MCMC), which was first adopted as a technique for performing approximate Bayesian inference by Neal (1993). Markov chains can be used to generate samples from a model's posterior, which would be difficult to sample from directly. In the limit of infinite samples, they draw samples from the true posterior distribution (Farquhar, 2022). MCMC-based approaches have been developed to scale to large datasets via, e.g., stochastic gradient MCMC (Welling and Teh, 2011) and Hamiltonian Monte Carlo (Neal, 1996; Izmailov et al., 2021b).

In contrast, Variational Inference (VI) approximates the posterior via a variational distribution. Following Hinton and Van Camp (1993), the approach was further developed by Graves (2011) and optimized by Blundell et al. (2015), who placed a mixture of Gaussian priors over each weight and optimized each mixture component. This led to improved performance but limited the application of the method to complex models as it doubled the number of parameters (Gal, 2016).

Laplace approximation was used by MacKay (1992) to perform approximate inference in Bayesian neural networks by fitting a Gaussian distribution centered at the networks' weight parameters after training, using the inverse of the Hessian of the loss function at these parameters to estimate the covariance. A different method applied after training is Gaussian stochastic weight averaging, introduced by Maddox et al. (2019), which fits a Gaussian distribution over the weight space based on the trajectory of stochastic gradient descent to capture the posterior distribution of the weights.

In contrast, other techniques aim to approximate the posterior predictive distribution, specifically the effect of uncertainty in the weights on the predictions, without explicitly computing the distributions over the weights. This group notably includes Monte Carlo (MC) dropout (Gal and Ghahramani, 2016) and deep ensembles (Lakshminarayanan et al., 2017). Although these methods were historically classified as non-Bayesian, they have been demonstrated to approximate ideal Bayesian inference more closely than many other approaches (Wilson, 2021).

Presenting MC dropout, Gal and Ghahramani (2016) showed that dropout can be viewed as VI with Bernoulli priors over the weights, simplifying the approximation compared to fully factored Gaussians by using Bernoulli noise in the hidden units (Farquhar, 2022). Importantly, MC dropout is model-agnostic and can be applied to any model with dropout regularization. However, the method requires multiple stochastic forward passes to perform approximate Bayesian inference, adding a significant computational overhead.

Lakshminarayanan et al. (2017) introduced deep ensembles as a simple and scalable alternative to Bayesian neural networks. Deep ensembles combine several neural networks sharing the same architecture trained independently on the same data with different initializations. They quantify uncertainty by analyzing the variance across the ensemble predictions. Lakshminarayanan et al. (2017) offered an ensemble interpretation of MC dropout, where predictions are averaged over an ensemble of neural networks that share parameters. Notably, this technique can be used with models that do not employ dropout regularization, making it another versatile, model-agnostic approach for uncertainty

estimation. However, training and saving multiple neural networks can be costly in terms of computational resources and storage.

Traditional methods for performing approximate Bayesian inference require expensive sampling. Thus, they have a significant computational overhead, making them less suitable for scenarios where we desire real-time information about a model's confidence in its predictions. Existing sampling-free uncertainty quantification methods include temperature scaling (Guo et al., 2017), where the softmax function is modified by introducing a temperature parameter to transform its uncalibrated into calibrated outputs by adjusting their sharpness, and variance propagation (Postels et al., 2019), where the variance of the input is propagated through the network to estimate the uncertainty of the outputs without requiring multiple forward passes or explicit sampling. Recent work has focused on distilling the uncertainty information from a probabilistic teacher model into a deterministic student to provide sampling-free uncertainty estimates (Shen et al., 2021).

In a similar vein, Bucila et al. (2006) sought to transfer the superior performance of large and slow neural models onto small and fast networks that comply with the time and space requirements imposed on models in many practical applications. For this purpose, they introduced model compression, where the functional mapping of a complex model is compressed into a substantially smaller model while retaining predictive performance. Similarly, Hinton et al. (2015) proposed knowledge distillation to mitigate the computational cost induced by MC and ensemble methods. Knowledge distillation involves training a performant student on the outputs of a large and complex model. Specifically, Hinton et al.'s (2015) approach uses a teacher that performs approximate Bayesian inference and distills the mean of its predictive posterior distribution into a student network. Applying knowledge distillation to an ensemble typically means that the ensemble's diversity is lost. The distribution distillation method introduced by Malinin et al. (2019) particularly aimed to distill the entire distribution of an ensemble. To incorporate further information about the teacher's uncertainty into the distillation process, Shen et al. (2021) proposed uncertainty-aware distribution distillation. Showing that their method can distill epistemic and aleatoric uncertainty, they also demonstrated its ability to enable real-time uncertainty quantification.

Meanwhile, the introduction of the transformer model by Vaswani et al. (2017) revolutionized NLP by replacing recurrent layers with self-attention mechanisms, allowing models to focus on different parts of the input sequence when predicting a part of the sequence and enabling them to process data in parallel. Devlin et al. (2019) further refined this architecture and introduced BERT, a bidirectional transformer used for various NLU tasks, which has successively become one of the most widely utilized pre-trained language models (Minaee et al., 2021). Subsequent research has explored several strategies to improve the robustness and performance of BERT: RoBERTa (Liu et al., 2019) performed training with less data and modified the pre-training process to enhance robustness, ELECTRA (Clark et al., 2020) introduced a different pre-training task to improve efficiency, and DeBERTa (He et al., 2020) achieved state-of-the-art results by modifying the self-attention mechanism. In parallel, Radford et al. (2018) developed OpenGPT, focusing on left-to-right transformers for natural language generation. Sanh et al. (2019) successfully applied model compression to BERT using knowledge distillation, retaining close to 99 percent of performance while featuring 40 percent fewer parameters and performing inference 60 percent faster. Similarly, Mukherjee and Awadallah (2020) distilled BERT into a BiLSTM student using an unlabeled transfer dataset in combination with a labeled dataset. Transformers have been used in conjunction with the Bayesian framework, whose application has focused on the specifics of the transformer architecture. Shelmanov et al. (2021) applied MC dropout to the ELECTRA model, modifying the standard MC dropout approach to account for neuron correlations. This adaptation improves prediction diversity and reduces computational costs compared to the traditional MC dropout method. Furthermore, Bahuleyan et al. (2018) explored VI techniques, specifically through Variational Attention, where either the entire network or layer subsets are treated with a variational approach.

Uncertainty quantification is essential for applications where mistakes can have severe implications, such as hate speech detection, ensuring that harmful content is accurately identified without wrongly censoring legitimate speech (Miok et al., 2022). Varshney and Alemzadeh (2017) motivated the use of uncertainty quantification for safety-critical machine learning tasks and advocated for the incorporation of epistemic uncertainty to enhance safety. Previous works such as Pruksachatkun et al. (2021) highlighted its relevance for NLU tasks, specifically in the social media domain, while Miok et al. (2022) further motivated the use of uncertainty quantification for reliable hate speech detection. Several works applied deep learning to text classification (Georgakopoulos et al., 2018; Mozafari et al., 2020) and investigated the application of uncertainty quantification to neural networks for text classification (He et al., 2020; Hu and Khan, 2021; Van Landeghem et al., 2022). Readily available pre-trained language models are usually not calibrated well for uncertainty, especially for OOD data (Guo et al., 2021). Van Landeghem et al. (2022) conducted a comparative analysis of the performance and calibration of popular uncertainty quantification methods, including MC dropout and deep ensembles applied to convolutional neural networks and BERT, evaluating them under different scenarios of distribution shift. Investigating uncertainty estimation for hate speech detection, Miok et al. (2022) found MC dropout to work well when applied to BERT, resulting in reliable uncertainty estimates.

# 3 Theory

This section provides a comprehensive overview of core theoretical concepts related to uncertainty quantification in deep learning models. In addition, it outlines the conceptual foundations of uncertainty-aware distribution distillation.

## 3.1 Uncertainty in Deep Learning

We can categorize predictive uncertainty into aleatoric and epistemic uncertainty (Hüllermeier and Waegeman, 2021). Almost any data-generating process encountered in practice features a stochastic component that is irreducible regardless of the amount of available data. Any input-output mapping formalized by a model is thus non-deterministic. Aleatoric uncertainty (AU) refers to this irreducible part of uncertainty.

We define a dataset $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{Y}) = (\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1}^N$, where each $(\boldsymbol{x}_i, \boldsymbol{y}_i) \in (\mathcal{X} \times \mathcal{Y})$. We let $f_{\boldsymbol{w}}(\boldsymbol{x})$ be a neural network such that $f$ is the mapping from the feature space $\mathcal{X}$ to the label space $\mathcal{Y}$, $f : \mathcal{X} \mapsto \mathcal{Y}$, with $\boldsymbol{w} = \{W_i\}_{i=1}^L$ the parameters of the network with $L$ layers. The prediction of a model is represented as a probability distribution $P(\boldsymbol{y}_i|\boldsymbol{x}_i)$ over possible outcomes $\boldsymbol{y}_i$ for a given input $\boldsymbol{x}_i$, which captures the inherent noise or variability in the data. In the context of regression, we typically assume that the model outputs are normally distributed with a mean $\mu(\boldsymbol{x})$ and variance $\sigma^2(\boldsymbol{x})$ that are functions of $\boldsymbol{x}$, where the variance captures the aleatoric uncertainty. Aleatoric uncertainty can be homoscedastic or heteroscedastic. Homoscedastic aleatoric uncertainty assumes that the observation noise is constant for different inputs, i.e., $\boldsymbol{y}_i = f_{\boldsymbol{w}}(\boldsymbol{x}_i) + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, with $f(\boldsymbol{x}_i)$ the deterministic part of the prediction of the model for input $\boldsymbol{x}_i$ and $\epsilon_i$ a noise term for prediction $i$. We speak of heteroscedastic aleatoric uncertainty if the observation noise is varying and input-dependent. Modeling this type of uncertainty is particularly useful if some parts of the input space are likely to have higher noise levels than others. Formally, $\boldsymbol{x}_i = f_{\boldsymbol{w}}(\boldsymbol{x}_i) + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, \sigma^2(\boldsymbol{x}_i))$.

Contrary to aleatoric uncertainty, epistemic uncertainty (EU) is not rooted in the data-generating process's inherent randomness but in the lack of knowledge about the optimal predictor. This lack of knowledge can refer to both the model structure and its parameters. Exposing a predictor to more data can fill its knowledge gaps. Hence, epistemic uncertainty is, in principle, reducible with the availability of more data. We can thus define the total uncertainty (TU) as $TU = AU + EU$.

In deep neural networks, we make no explicit assumption about the global structure that governs how inputs relate to outputs (Hüllermeier and Waegeman, 2021). Any class is approximated locally by the region in the input space of samples belonging to that class. Aleatoric uncertainty arises in areas where these regions intersect, indicating inherent data variability or noise. Meanwhile, epistemic uncertainty occurs in areas where no examples have been seen so far.

Kendall and Gal (2017) highlight that even in scenarios where enough data is available to explain away the epistemic uncertainty, it is nevertheless required to capture this uncertainty to detect inputs that the model has not previously encountered.

In summary, modeling aleatoric uncertainty is essential for large data applications where the epistemic uncertainty is explained away and real-time applications as no sampling is required to generate uncertainty estimates. On the other hand, modeling epistemic uncertainty is necessary for safety-critical applications where exposure to OOD data is likely and small datasets where training data is sparse (Kendall and Gal, 2017).

## 3.2 Bayesian Deep Learning

Uncertainty is inherently connected to the Bayesian paradigm. Via Bayes' theorem, we define the posterior distribution over the model parameters $\boldsymbol{w}$ given a dataset $(\boldsymbol{X}, \boldsymbol{Y})$ as

$$p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{Y}) = \frac{p(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{w})p(\boldsymbol{w})}{p(\boldsymbol{Y}|\boldsymbol{X})},$$

where $p(\boldsymbol{w})$ represents a prior distribution over the hypothesis space, captured by the model parameters (Hüllermeier and Waegeman, 2021). The hypothesis space is defined as all possible parameter settings that could "explain" the input-output mapping. We learn the distribution of the unknown parameters over this space by updating our prior knowledge with the observed data to obtain the posterior.

We find the predictive distribution of an output $\boldsymbol{y}_i$ for an input $\boldsymbol{x}_i$ as the posterior of the data likelihood

$$p(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{X}, \boldsymbol{Y}) = \int p(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{w})p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{Y})d\boldsymbol{w}, \tag{1}$$

where we need to integrate over the parameters $\boldsymbol{w}$ to obtain the predictive distribution (Gal, 2016). This predictive distribution $p(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{X}, \boldsymbol{Y})$ represents a Bayesian model average, i.e., it considers all possible hypotheses, weighted by their posterior probabilities (Wilson and Izmailov, 2020).

Variability in this posterior predictive distribution captures the epistemic uncertainty (Hüllermeier and Waegeman, 2021). In short, epistemic uncertainty is the uncertainty about the model parameters.

In cases when we cannot compute the integral in closed form, we could compute Equation 1 using an MC approximation

$$p(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{X}, \boldsymbol{Y}) \approx \frac{1}{J} \sum_{j=1}^{J} p(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{w}_j),$$

with

$$\boldsymbol{w}_j \sim p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{Y})$$

if we knew the posterior distribution of the parameter space $\boldsymbol{w}$ (Wilson and Izmailov, 2020). However, the weight posteriors $p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{Y})$ are intractable, even for small datasets and models (Bishop, 2006). Thus, we require approximate techniques. We can differentiate methods for approximating the posterior of the model parameters into local approximation methods, such as VI and Laplace approximation, and sampling-based methods, including Hamiltonian Monte Carlo and stochastic gradient Langevin dynamics. Approaches such as deep ensembles and MC dropout simulate sampling from the posterior and are used to directly approximate the distribution over the outputs, enabling uncertainty estimation.

The Bayesian paradigm can be applied to deep learning models, offering a probabilistic interpretation of neural networks (Gal, 2016). In Bayesian neural networks, we place a prior distribution over the network weights, typically a standard Gaussian distribution.

### 3.3 Approximate Bayesian Inference

In this subsection, we introduce common approaches for performing approximate Bayesian inference in Bayesian neural networks.

#### 3.3.1 Markov Chain Monte Carlo

MCMC methods are well-established and serve as the foundation of most modern sampling-based methods for approximating the posterior in Bayesian models. They are highly flexible and do not make assumptions about the form of the underlying distribution (Neal, 1996). In MCMC, an ergodic Markov chain, which eventually explores all regions of the target distribution, generates a series of dependent samples. Since we define the Markov chain so that its stationary state corresponds to the true predictive posterior, we are guaranteed to sample from the true posterior asymptotically. Despite the inherent dependence, the estimate of the interval converges to the true value with an increasing number of samples (Neal, 1996). However, due to these dependencies, the required number of samples may be large, and during the sampling process, the Markov chain may exhibit random walk behavior. Furthermore, a long warm-up period may be necessary before the equilibrium distribution of the chain is reached.

These drawbacks mean that the MCMC algorithm struggles in settings where datasets are large, or models are very complex (Blei et al., 2017). More efficient extensions of standard MCMC, such as Hamiltonian Monte Carlo and stochastic gradient Langevin dynamics, aim to overcome these limitations and make MC methods applicable to high-dimensional distributions and complex models.

#### 3.3.2 Variational Inference

A different approach is to minimize the Kullback-Leibler (KL) divergence between a variational distribution $q_{\boldsymbol{\theta}}(\boldsymbol{w})$ over the weights, from which we can easily sample, and the true posterior distribution with respect to $\boldsymbol{\theta}$,

$$\mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{w})||p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{Y})) \int q_{\boldsymbol{\theta}}(\boldsymbol{w}) \log \frac{q_{\boldsymbol{\theta}}(\boldsymbol{w})}{p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{Y})} d\boldsymbol{w}.$$

Since we cannot access the true posterior distribution, the KL divergence is not directly tractable (Farquhar, 2022). We instead maximize the evidence lower bound (ELBO),

$$\mathrm{ELBO}(q_{\boldsymbol{\theta}}, p) = \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{w})}[\log p(\boldsymbol{X}, \boldsymbol{Y}|\boldsymbol{w})] - \mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{w})||p(\boldsymbol{w})),$$

which is equivalent to minimizing the KL divergence. Here, we find $q^*_{\boldsymbol{\theta}}(\boldsymbol{w})$ as the minimum of this optimization objective. In this process, optimization thus replaces marginalization (Gal, 2016). We can then approximate the predictive distribution as

$$p(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{X}, \boldsymbol{Y}) \approx \int p(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{w}) q^*_\theta(\boldsymbol{w}) d\boldsymbol{w}.$$

As an alternative to MCMC methods, VI tends to be easier to scale to large data (Blei et al., 2017). Compared to MCMC, it only approximates the true density, with the benefit of being faster. This makes it an alternative for modern larger-scale neural networks, where MCMC sampling is not feasible.

#### 3.3.3 Deep Ensembles

Deep ensembles represent a practical alternative for performing approximate Bayesian inference (Lakshminarayanan et al., 2017). In its simplest form, a deep ensemble is the same neural network trained multiple times on the same data. The predictions of the resulting models are averaged to capture uncertainty, thus providing a robust method for uncertainty quantification. Deep ensembles are highly flexible and can handle high-dimensional parameter spaces, as well as non-Gaussian and multi-modal posteriors (Wilson and Izmailov, 2020).

By employing multiple training runs with different initializations, deep ensembles explore different low-loss regions within diverse basins of attraction, capturing typical points representing significant posterior mass (Wilson and Izmailov, 2020). This approach enables ensembles to represent the entire posterior landscape and increases their functional diversity. The averaging of the outputs of these different models effectively implements Bayesian model averaging through a form of practical MC integration. This integration allows deep ensembles to act as an effective mechanism for approximate Bayesian marginalization. We can view it as generating samples from an approximate posterior distribution, thus providing a good approximation to the Bayesian predictive distribution in complex, high-dimensional settings.

#### 3.3.4 Monte Carlo Dropout

Dropout is a stochastic regularization technique that introduces randomness into a model by selectively deactivating different pathways for the data through the network during each training batch. This is achieved by adding dropout layers to the network architecture, from which units are randomly dropped along with their associated connections with a set probability $p_i$ for a layer $i = 1, \ldots, L$ (Srivastava et al., 2014). At test time, dropout is deactivated, but the network weights are adjusted accordingly to obtain predictions. Specifically, the weights of a layer $i$ are scaled down by multiplying them by $1 - p_i$ to account for the average contribution of each node during training.

MC dropout interprets dropout as a form of VI in Bayesian neural networks. Here, we treat each weight as a random variable influenced by a Bernoulli-distributed dropout mask (Gal and Ghahramani, 2016). This representation allows dropout to mimic sampling from an approximate Bayesian posterior, enabling a probabilistic model interpretation. During the training phase, dropout serves as a regularizer, preventing overfitting by discouraging complex co-adaptations on the training data, equivalent to a Bayesian prior (Gal, 2016). MC dropout uses multiple stochastic forward passes with different dropout masks to obtain predictions, averaging the results to approximate the Bayesian posterior mean. Individual samples correspond to samples from the approximate posterior predictive distribution. Hence, this method enhances generalization by regularizing the network and enables the straightforward estimation of predictive uncertainty.

### 3.4 Combining Aleatoric and Epistemic Uncertainty

Bayesian deep learning approaches tend to capture epistemic or aleatoric uncertainty, but not both (Gal, 2016). As explained above, they can be modeled by a probability distribution over the model parameters or the model outputs, respectively. Modeling aleatoric uncertainty alone is computationally efficient, as the observation noise can be learned as a function of the data via a special loss function (Kendall and Gal, 2017). However, this comes at the cost of not being able to effectively identify OOD examples, for which modeling epistemic uncertainty is better suited. The remedy is a unified Bayesian deep learning framework, where we learn the mapping from the inputs to the aleatoric uncertainty, combined with an approximation of the epistemic uncertainty. For this purpose, Kendall and Gal (2017) suggest placing a distribution over the weights to capture the epistemic uncertainty and approximating the posterior over the Bayesian neural network using the dropout variational distribution, i.e., MC dropout. Additionally, they map the input to the target prediction and the observation noise in the form of the variance to measure the aleatoric uncertainty. Hence, the final model output comprises the predictive mean and the variance. For regression, we can assume that the model output is normally distributed.

To adapt the technique to classification models, the heteroscedastic regression uncertainty is placed over the logit space instead (Kendall and Gal, 2017). When passed through a sigmoid or softmax function, the logits form a probability

vector for a binary or multiclass classification problem, respectively. Specifically, we assume normally distributed logits and hence place a Gaussian distribution over the logit output.

Given a neural network $f_{\boldsymbol{w}}(\boldsymbol{x})$, $p(\boldsymbol{y}|f_{\boldsymbol{w}}(\boldsymbol{x})) = \mathrm{Softmax}(f_{\boldsymbol{w}}(\boldsymbol{x}))$ yields the probability vector. We now modify the network so that it instead outputs the input-dependent observation noise along with the mean prediction as $[\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}}^2] = f_{\boldsymbol{w}}(\boldsymbol{x})$. In practice, we instead let the network predict the log variance since it is more numerically stable than predicting the variance by avoiding potential divisions by zero. In summary, we obtain a prediction for an input $\boldsymbol{x}_i$ as $\hat{\boldsymbol{y}}_i \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\sigma}}_i^2)$, where the logit output $\hat{\boldsymbol{\mu}}_i$ is perturbed with Gaussian noise with the variance $\hat{\boldsymbol{\sigma}}_i^2$. Then, we apply the softmax operation to the corrupted logit output vector $\hat{\boldsymbol{y}}_i$ to get the associated probability vector $\hat{\boldsymbol{p}}_i = \mathrm{Softmax}(\hat{\boldsymbol{y}}_i)$. The expected log-likelihood for this model corresponds to $\log \mathbb{E}_{\mathcal{N}(\hat{\boldsymbol{y}}_i;\hat{\boldsymbol{\mu}}_i,\hat{\boldsymbol{\sigma}}_i^2)}[\hat{\boldsymbol{p}}_{i,c}]$, with $c$ the observed class for input $i$. Since we cannot analytically integrate out the Gaussian distribution, we perform MC integration to approximate the objective. This process is optimized as we only pass the inputs once through the model to obtain the logit output alongside the variance. We then sample from the distribution parametrized by the predicted mean and variance. Therefore, the sampling process does not result in significant compute overhead. The same procedure is used to generate uncertainty estimates, implying that we can efficiently quantify uncertainty at test time. Rewriting the above, we obtain the corrupted logits over the $t = 1, \ldots, T$ MC samples as $\hat{\boldsymbol{y}}_{i,t} = \hat{\boldsymbol{\mu}}_i + \hat{\boldsymbol{\sigma}}_i^2 \epsilon_t, \epsilon_t \sim \mathcal{N}(0, I)$. Then, the numerally stable stochastic loss function is

$$\mathcal{L}_y = \sum_i \log \frac{1}{T} \sum_t \exp(\hat{y}_{i,t,c} - \log \sum_{c'} \exp \hat{y}_{i,t,c'}), \tag{2}$$

with $y_{i,t,c'}$ the $c'$ element in the logit vector $y_{i,t}$ (Kendall and Gal, 2017). We can interpret this objective as learning loss attenuation, where, by adding uncertainty to the network output, the model gains a self-correcting mechanism that reduces the impact of erroneous labels throughout the learning process. This makes predictions less confident, which improves the robustness against noisy data. However, the model incurs a penalty for excessive uncertainty, preventing it from ignoring the data entirely. This balance results directly from the model's probabilistic foundation, ensuring it remains attentive to the data while managing the uncertainties.

### 3.5 Distillation

Model compression (Bucila et al., 2006) aims to transfer the capabilities of a large, complex model into a smaller, more efficient one without substantial performance loss. It involves training a compact neural network to mimic the function of a more cumbersome model, specifically an ensemble. This is achieved by first using the ensemble to label a large set of unlabeled data, creating a new training dataset, typically much larger than the training data used to train the ensemble. The smaller model is then trained on this dataset, effectively compressing the original model's knowledge into a form that is easier to deploy in resource-limited settings.

Hinton et al. (2015) expand model compression to the concept of knowledge distillation, focusing on the generalization ability of the complex model. Knowledge distillation aims to distill the knowledge of a model ensemble into a single model, where "knowledge" refers to the learned mapping from the input vectors to the output vectors, $\mathcal{X} \mapsto \mathcal{Y}$. This process hinges on the assumption that a large teacher model, particularly an ensemble of diverse models, often exhibits superior generalization. A small student model trained to learn this generalization will do much better on test data than if trained simply on the same training data as the ensemble. To transfer the ensemble's generalization ability, knowledge distillation centers on using the class probability outputs from the teacher model as "soft targets" for training the student model. These can be generated on the ensemble's original training data or a disparate transfer dataset. Soft targets substantially facilitate the distillation process by providing more information than hard targets, the class labels, especially when they exhibit high entropy. This allows the student model to be trained on less data and with a much higher learning rate than the teacher. In cases where the large model is consistently extremely confident about its predictions, much of the information about the learned input-output mapping resides in the very small output probabilities. However, their influence on the transfer loss function would be negligible. Hinton et al. (2015) propose a solution for this problem where the temperature of the softmax is raised to make the teacher predict suitably soft probabilities. In the softmax layer, the logits $z_c$ for each class are converted into probabilities $q_c$ by comparing $z_c$ with the other logits

$$q_c = \frac{exp(z_c/T)}{\sum_{c'} exp(z_{c'}/T)},$$

where $T$ refers to the temperature. Normally, $T$ is set to 1, but increasing it results in a softer class probability distribution, thus emphasizing the information in the small probabilities. To further improve the distillation process, Hinton et al. (2015) suggest using the soft targets alongside the ground truth labels. Hence, in its simplest form, knowledge distillation involves training the student model on the transfer set, where each transfer training case is produced using the teacher model with a high softmax temperature. During training, the student uses the same high temperature, which is then reset to 1 once trained to generate predictions. Incorporating the ground truth labels in

the objective function further improves knowledge distillation. The distillation loss is then a weighted average of the cross-entropy between the true labels and the student outputs with the softmax temperature set to 1 and the cross-entropy between the soft targets and the student outputs with high softmax temperature, the same as was used in the teacher for generating the soft labels.

Building directly on Hinton et al. (2015), Shen et al. (2021) propose a probabilistic distillation framework wherein a deterministic student learns to approximate the posterior predictive distribution of the Bayesian teacher. Specifically, a deterministic neural network student $f_{\boldsymbol{\phi}}(\boldsymbol{x})$ learns to output the parameters of a variational distribution $r(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D}_{train})$, which approximates the predictive distribution $q(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D}_{train})$ of an MC dropout teacher, $f_{\boldsymbol{w}}(\boldsymbol{x})$, with $\mathcal{D}_{train} = (X_{train}, Y_{train})$. We can then use the student network to obtain reliable uncertainty estimates in a single forward pass, eliminating the MC dropout sampling overhead. The framework is able to quantify aleatoric in addition to epistemic uncertainty by outputting an additional parameter to capture the input-dependent observation noise together with the prediction. The teacher's output then becomes $[\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}}^2] = f_{\boldsymbol{w}}(\boldsymbol{x})$, where $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\sigma}}^2$ correspond to the predictive mean and the observation noise, respectively (Shen et al., 2021; Kendall and Gal, 2017). In summary, we apply the teacher-student paradigm to train the student network $f_{\boldsymbol{\phi}}(\boldsymbol{x})$ as follows: The Bayesian teacher $f_{\boldsymbol{w}}(\boldsymbol{x})$, an MC dropout model, is trained on the data $\mathcal{D}_{train}$. In the second step, we then generate samples from $f_{\boldsymbol{w}}(\boldsymbol{x})$ that serve as observations from the approximate posterior predictive distribution $q(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D}_{train})$. Following the argument of Hinton et al. (2015), we sample the logits instead of the class probabilities to better capture the information contained in the teacher outputs. Finally, we train the student $f_{\boldsymbol{\phi}}(\boldsymbol{x})$ on these samples to learn the mapping from the inputs to the parameters of $r(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D}_{train})$, the student's approximation of the teacher's posterior. In this training process, matching the logits of the teacher model corresponds to a special case of distillation (Hinton et al., 2015).

# 4 Setup

The following section describes in detail the data, model, and evaluation metrics used throughout this study.

## 4.1 Data

We conduct our experiment using a popular text classification dataset first introduced by Davidson et al. (2017). The data comprise labeled posts on the Twitter/X social media platform, which have been assigned to three different categories, namely hate speech, offensive language, and neither[1]. For the sake of our analysis, we transform the multiclass into a binary classification problem by grouping all text sequences considered hate speech or offensive language into a single category.

Some preprocessing steps are required before we can use the data in our analysis. These steps are partially taken from Mozafari et al. (2020), who use the same dataset in their study of hate speech classification with BERT. Besides basic text cleaning, we encode selected text entities with tags, some of which are exclusive to Twitter/X. Specifically, we encode all user mentions as "<user>," all numbers as "<number>," all hashtags as "<hashtag>," all URLs as "<URL>," all emoticons as "<emoticon>," and all emojis as "<emoji>."

The 24,783 tweets contained in the dataset are in English, with a minimum, average, and maximum sequence length in the preprocessed dataset of 1, 14.5, and 43 tokens, respectively. Encoded entities comprise roughly ten percent of all tokens, most of which are user mentions and emojis. The dataset exhibits a noticeable class imbalance, as 83.2 percent of all sequences are classified as hate speech.

## 4.2 Model

Many of the most popular current deep learning architectures are transformer-based (Minaee et al., 2021). In the NLP domain, the most widely-used ones include BERT and its successors, such as RoBERTa and DeBERTa, as well as the family of Generative Pre-trained Transformers (cf. Section 2). Of all the options available, we restrict ourselves to one for this study. Since its introduction, BERT has established itself as a staple for text classification in theory and practice. It is a commonly used general-purpose NLP architecture and is frequently employed by other studies in the domain. Hence, due to its popularity, any results of our research will be of great interest to practitioners who employ BERT and could easily modify their existing approach to incorporate the proposed method. Furthermore, our findings have implications for newer BERT-based architectures. Our analysis uses the base variant of BERT for uncased input text. It features 12 transformer encoding layers and a total of 110 million parameters. Since BERT requires a specific input format, we must employ a tokenizer. Here, we use a universal maximum length of 48 since the maximum sequence

[1] Please refer to Davidson et al. (2017) for a detailed overview of the data collection and labeling process.

length is 43, as noted above. We pad all sequences to the maximum length. All following models are trained with the Adam optimizer (Kingma and Ba, 2014).

### 4.3 Evaluation Metrics

We use standard classification metrics to evaluate the predictive performance, namely accuracy, precision, recall, F1 score, AUC, and Brier score. As in Shen et al. (2021), we employ additional measures to evaluate the uncertainty estimate. The quality of uncertainty is typically measured through calibration (Kendall and Gal, 2017), in which a model is considered calibrated if its predictive probabilities align with the empirical frequencies observed in the data (Gal and Ghahramani, 2016).

A popular measure of model calibration is the expected calibration error (ECE), first defined by Guo et al. (2017) as

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right|,$$

where

$$\mathrm{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbb{1}(\arg\max_c \hat{y}_{i,c} = y_{i,c}),$$

with $\arg\max_c \hat{y}_{i,c}$ the predicted and $y_{i,c}$ the true class label for class $c$ of a sample $i$, and

$$\mathrm{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_{i,c},$$

with $\hat{p}_{i,c}$ the predicted probability for class $c$ of a sample $i$. For a perfectly calibrated model, $\mathrm{acc}(B_m) = \mathrm{conf}(B_m)$ for all bins $m \in 1, \ldots, M$, and hence $\hat{p}_{i,c}$ would represent a true probability. The ECE thus measures the calibration of a model, specifically the agreement between predicted probabilities and observed outcomes across different bins. A lower ECE suggests that the predicted probabilities are, on average, closer to the true outcome frequencies within each bin (Guo et al., 2017). Shen et al. (2021) compute the ECE using the L2 norm and a bin size of 30. However, more typical in the existing literature is the L1 norm in combination with a bin size of 10, as used, among others, by Ovadia et al. (2019). Hence, we opt to compute the ECE with the L1 norm and 10 bins in all the following analyses.

As an additional metric that is more specifically suited to quantify the uncertainty of the model in classification tasks, we use the mutual information between the predictions and the posterior model, as also done by Shen et al. (2021). The Bayesian Active Learning by Disagreement (BALD) score was introduced by Houlsby et al. (2011) to evaluate the predictive uncertainty of a model and is defined as

$$\mathrm{I}[y, \omega | x, \mathcal{D}_{\text{train}}] = \mathbb{H}[y | x, \mathcal{D}_{\text{train}}] - \mathbb{E}_{\omega}[\mathbb{H}[y | x, \omega]].$$

It computes the difference between the entropy of the predictive distribution and the expected entropy of the predictive distribution. For uncertainty quantification methods such as MC dropout, the predictive entropy is typically calculated as the entropy of the averaged predictions over multiple stochastic forward passes. In contrast, the expected entropy is the average of the entropy of the predictions of each stochastic forward pass.

## 5 Uncertainty-Aware Distribution Distillation

In the following section, we describe how to adapt the uncertainty-aware distribution distillation method, as introduced by Shen et al. (2021), for the BERT architecture in a text classification setting, specifically for the task of hate speech detection. This entails a step-by-step explanation of the distillation process, an experimental analysis of the optimal parameter settings for the teacher and student model, and a detailed comparison of teacher and student in terms of predictive performance and uncertainty quality. Holistically, this section addresses Research Question 1, which seeks to understand whether we can successfully distill a student model using the uncertainty distillation methodology. In addition, it aims to provide insights into the methodology's adaptation to the specifics of the BERT architecture in the context of an NLP problem setting.

### 5.1 Methodology

Uncertainty-aware distribution distillation aims to distill the conditional predictive distribution of a pre-trained dropout model (Shen et al., 2021). It features three stages: Training a cumbersome model in the form of an MC dropout teacher, generating a transfer dataset using predictive samples from the teacher, and training a student model on this transfer dataset.

### 5.1.1 Dropout Teacher Training

Shen et al. (2021) apply uncertainty distillation exclusively to the CV domain, namely, semantic segmentation and depth estimation. For this purpose, they employ two neural networks based on the convolutional neural network architecture, one with a decoder-encoder structure and another with a ResNet architecture. As uncertainty distillation focuses on distilling the information about the teacher's uncertainty into the student, without performing model compression, the students feature the same architecture as their respective teachers. Both teacher networks feature dropout layers as a part of their respective architecture, which are used to generate uncertainty estimates via MC dropout. In the students, these layers are disabled. Similarly, this study's teacher and student are built on the same BERT architecture. Since BERT natively features dropout layers inside its transformer blocks (Devlin et al., 2019), we do not add them explicitly.

We can incorporate aleatoric and epistemic uncertainty into the distillation process. As Kendall and Gal (2017) note, modeling both aleatoric and epistemic uncertainty together is essential in safety-critical real-time applications that deal with large amounts of data. This corresponds to the typical setting encountered for hate speech detection models in practice. Shen et al. (2021) show that this improves uncertainty performance overall. To quantify aleatoric uncertainty, we let the models output the observation noise together with the predictive mean output. Specifically, we introduce an additional output head for the predictive variance. In practice, we use the log variance instead of the variance to enable unconstrained optimization and avoid zero-division errors (Shen et al., 2021). Hence, the teacher output is $[\hat{\boldsymbol{\mu}}, \boldsymbol{s}] = f_{\boldsymbol{w}}(\boldsymbol{x})$, with $\boldsymbol{s} = \log(\hat{\boldsymbol{\sigma}}^2)$. Considering both model outputs, we train the teacher model using the loss function described in Equation 2, which we define as $\mathcal{L}_t$. To compute the final model performance and, more importantly, to quantify the model uncertainty, we use 50 MC dropout samples. Any results for the teacher model throughout the paper will be obtained this way unless stated otherwise.

Pre-trained language models such as BERT require fine-tuning on task-specific data to reach their full potential (Devlin et al., 2019). In this process, we additionally optimize several model hyperparameters to boost the model performance further. For this purpose, we partition the dataset into training, validation, and testing subsets, with 80 percent of the data allocated for training and 10 percent each for validation and testing. In the hyperparameter search, we tune the number of epochs, the learning rate, and the layer-wise dropout rates, which comprise the dropout probabilities for the attention and hidden layers inside the transformer blocks and the dropout layer placed before the output heads. We base the considered parameter ranges heavily on Devlin et al. (2019), while the ranges for the different dropout rates are loosely based on Van Landeghem et al. (2022). We perform a grid search over each possible hyperparameter combination. In each iteration, we fine-tune a model initialized with the pre-trained weights on the training set and evaluate this intermediate model on the validation set. Here, we use the F1 score to select the best-performing hyperparameter set. Once we have determined the best configuration, we fine-tune another model on the combined training and validation datasets and evaluate the resulting model on the test set to obtain the results. We consider learning rates of $2 \times 10^{-5}$, $3 \times 10^{-5}$, and $5 \times 10^{-5}$, and dropout rates of 0.2 and 0.3 for all dropout layers. As observed, varying the number of epochs within the range recommended by Devlin et al. (2019) does not significantly impact performance. Hence, we choose to keep it set to 3. The choice of hyperparameters appears to matter, but a large range of hyperparameter combinations yields a high F1 score. The learning rate seems to be the most critical factor, with the biggest impact on performance. We observe a significant drop in performance for higher learning rates. Within the specified range, the exact choice of dropout probabilities appears slightly less important, although we note a significant negative performance impact when considering more extensive ranges. Due to the complex interaction effects of the different hyperparameters, the sensitivity analysis is limited. Overall, we find a combination of a low learning rate of $2 \times 10^{-5}$, together with comparatively low but varying dropout rates set to 0.2 for hidden dropout, 0.2 for attention dropout, and 0.3 for classifier dropout, trained for 3 epochs, to yield the best performance, as measured by the F1 score. The resulting teacher model obtains an F1 score and AUC of 0.98 and 0.992 and an ECE and Brier score of 0.022 and 0.024, respectively.

### 5.1.2 Transfer Dataset

Now that we have obtained a suitable teacher model, we can proceed with the second stage of the distillation process: generating a transfer dataset. As noted previously, we can greatly enhance the distillation result by using soft instead of hard targets to train the student model. We obtain the soft targets for the transfer dataset by sampling from the teacher's predictive distribution, which approximates the true posterior. This dataset can be either the same dataset the teacher was trained on or a separate one. This transfer dataset needs to be labeled for additionally leveraging ground truth points, significantly improving the predictive performance (Hinton et al., 2015; Shen et al., 2021).

In uncertainty distillation, the teacher samples can incorporate either exclusively epistemic or a combination of aleatoric and epistemic uncertainty. Shen et al. (2021) note that the student may underestimate the teacher's epistemic uncertainty when trained on the same dataset as the teacher due to overfitting of the teacher model. Hence, they conclude that there should ideally be no overlap between the training data of the teacher and student models. As labeled data can be

scarce and expensive, they employ extra image data augmentations not used to augment the teacher training data to perturb the student training set so that the original training dataset can still be used for both models. Shen et al. (2021) show that these extra augmentations of the student training dataset can significantly enhance the quality of uncertainty estimates. They note that this boost does not result directly from the augmentations but from the augmented teacher samples that better represent the test-time predictive distribution. We hypothesize that, unlike the models discussed by Shen et al. (2021), which do not use pre-trained weights, a pre-trained language model such as BERT might already possess sufficient generalization capability. Therefore, data augmentations may not be required for BERT. We will explore this hypothesis further in subsequent analyses.

**Algorithm 1** Generating Uncertainty-Distorted Samples From the Bayesian Teacher

**Inputs:**
  Input data $\boldsymbol{x}$, number of predictive samples $m$, number of random samples $k$
**Output:**
  Distorted teacher samples $\hat{\boldsymbol{y}}_{tl}$, for $t = 1, \ldots, m$ and $l = 1, \ldots, k$
**for** $t = 1$ to $m$ **do**
  Generate teacher MC dropout samples $[\hat{\boldsymbol{\mu}}_t, \hat{\sigma}_t^2] = f_{\boldsymbol{w}_t}(\boldsymbol{x})$
  Compute empirical mean observation noise $\tilde{\sigma}^2 \triangleq \frac{1}{m}\sum_{t=1}^{m} \hat{\sigma}_t^2$
  Obtain stabilized teacher samples as $[\hat{\boldsymbol{\mu}}_t, \tilde{\sigma}^2]$, $t = 1, \ldots, m$
  **for** $l = 1$ to $k$ **do**
    Generate samples from standard normal distribution $\epsilon_l \sim \mathcal{N}(0, 1)$
    Obtain final distorted predictive samples $\hat{\boldsymbol{y}}_{tl} = \hat{\boldsymbol{\mu}}_t + \tilde{\sigma}^2 \epsilon_l$
  **end for**
**end for**

If we care exclusively about incorporating epistemic uncertainty, then sampling from the teacher is straightforward: We generate $m$ predictive samples from the teacher for each input in the transfer training set, corresponding to the teacher's original training dataset. In practice, this corresponds to simple MC dropout sampling, where we obtain $m$ samples from the approximate posterior for each sequence in the transfer dataset. This process becomes more involved if we desire to model both aleatoric and epistemic uncertainty, as outlined in Algorithm 1. As before, we generate $m$ teacher samples $\{[\hat{\boldsymbol{\mu}}_t, \hat{\sigma}_t^2] = f_{\boldsymbol{w}_t}(\boldsymbol{x})\}_{t=1}^{m}$ for each input sequence to capture the epistemic uncertainty. We will use the observation noise to additionally model aleatoric uncertainty. To this end, we stabilize the sampling and subsequent training process by computing the empirical mean-variance across the $m$ samples as $\tilde{\sigma}^2 \triangleq \frac{1}{m}\sum_{t=1}^{m} \hat{\sigma}_t^2$, yielding the stabilized teacher samples $\{[\hat{\boldsymbol{\mu}}_t, \tilde{\sigma}^2]\}_{t=1}^{m}$. We then obtain $k$ random samples $\epsilon_l \sim \mathcal{N}(0, 1)$, $l = 1, \ldots, k$ from the standard normal distribution for each of the $m$ predictive samples for each input sequence to model the aleatoric uncertainty. We compute the distorted predictive samples as $\hat{\boldsymbol{y}}_{tl} = \hat{\boldsymbol{\mu}}_t + \tilde{\sigma}^2 \epsilon_l$, for $t = 1, \ldots, m$ and $l = 1, \ldots, k$, which contain both aleatoric and epistemic uncertainty. Shen et al. (2021) show in experiments that a small $m$ and $k$ are sufficient to perform the subsequent student distillation successfully. Hence, we use an identically small $m$ of 5 and $k$ of 10. Shen et al. (2021) perform the sampling process on-the-fly during training. Since our dataset is smaller, we perform the sampling after the training process.

### 5.1.3 Student Training

Having generated samples from the teacher model, we are ready to turn to training the student model next.

The student model is based on the same architecture as the teacher, again featuring an additional output head to account for the observation noise. However, the student does not feature dropout layers, as in Shen et al. (2021). Again, since the teacher and student comprise the same number of parameters, we do not perform model compression.

In the distillation process, the student $f_{\boldsymbol{\phi}}(\boldsymbol{x})$ learns to approximate the teacher $f_{\boldsymbol{w}}(\boldsymbol{x})$. We optimize $f_{\boldsymbol{\phi}}(\boldsymbol{x})$ using maximum likelihood estimation (MLE). Specifically, given the distorted teacher samples $\hat{\boldsymbol{y}}_{tl}$, for $t = 1, \ldots, m$ and $l = 1, \ldots, k$, we minimize the negative log-likelihood for each input $\boldsymbol{x}$ as

$$\mathcal{L}_s = -\sum_{t,l} \log r(\hat{\boldsymbol{y}}_{tl} | \boldsymbol{x}; \boldsymbol{\phi}),$$

where $r(\hat{\boldsymbol{y}}_{tl} | \boldsymbol{x}; \boldsymbol{\phi})$ is parametrized by $f_{\boldsymbol{\phi}}(\boldsymbol{x})$ Shen et al. (2021). The teacher's approximate predictive distribution $q(\boldsymbol{y} | \boldsymbol{x}, \mathcal{D}_{train})$ is modeled using a logit-normal distribution. Consequently, the student outputs $\boldsymbol{\mu}_i'$ and $s_i'$ represent the mean and log variance of the corresponding Gaussian distribution for an input $i$. A numerically stable Gaussian MLE

training objective can be derived as

$$\mathcal{L}_s = \frac{1}{N}\frac{1}{M}\frac{1}{K}\sum_{i,t,l}\frac{1}{2}\exp(-s_i')||\hat{\boldsymbol{y}}_{itl} - \boldsymbol{\mu}_i'||^2 + \frac{1}{2}s_i', \tag{3}$$

where $\hat{\boldsymbol{y}}_{itl}$ are the predicted logits sampled from the teacher, with added aleatoric distortion. Shen et al. (2021) observe empirically that training with this loss function alone leads to sub-optimal performance, possibly due to a noisy signal in the generated samples. Therefore, they employ ground truth labels alongside the soft teacher samples to stabilize the student training process. Using the loss function $\mathcal{L}_t$ used to train the teacher in combination with $\mathcal{L}_s$, the total loss becomes

$$\mathcal{L}_{total} = \mathcal{L}_s + \lambda\mathcal{L}_t, \tag{4}$$

where $\lambda$ is a tuneable hyperparameter. $\mathcal{L}_{total}$ is thus a combination of the binary cross-entropy and Gaussian MLE loss. We obtain uncertainty estimates by performing MC sampling on the logit space since no closed-form solutions exist for the moments of the logit-normal distribution. Sampling results in a negligible computational overhead during inference, as it essentially equals performing multiple forward passes through the output layer only. As for the teacher, we use 50 such samples to compute uncertainty estimates. We compute the predictive performance of the student using the sample mean.

To optimize the student performance, we again perform hyperparameter optimization via a grid search. Although we initialize the student with the teacher weights to achieve faster convergence, as proposed by Shen et al. (2021), we need to tune the student hyperparameters as the training dataset differs from that of the teacher and to account for the lack of dropout layers. In the grid search, we repeatedly fine-tune the student on the transfer dataset using different hyperparameter configurations and the loss function defined in Equation 4. In each iteration, we evaluate the model on the test set. We use the F1 score to determine the best model. Among the considered hyperparameters are the learning rate and the number of training epochs. We additionally tune the weight $\lambda$ in $\mathcal{L}_{total}$ to find the optimal balance of the ground truth and soft targets, as suggested by Shen et al. (2021), who find $\lambda = 1$ to perform well across their experiments. Shen et al. (2021) consistently opt for a lower student learning rate than the one used for the teacher, which is sensible given the student's significantly larger training dataset and its initialization with the teacher weights. Similarly, Shen et al. (2021) train the student for fewer steps than the teacher. Consequently, we consider loss weights $\lambda$ of 0.5, 1, and 2, learning rates of $2 \times 10^{-4}$, $2 \times 10^{-5}$, and $2 \times 10^{-6}$, and 2, 3, and 4 training epochs, also based on our settings for the teacher. Many of the resulting configurations yield a similarly high F1 score. Generally, the performance tends to be better when the student is trained for more epochs and with a higher learning rate. We do not observe a clear trend for the loss weight parameter. The dynamic between the different hyperparameters is again complex, limiting the analysis of their individual influence. To choose the final configuration of students, we take into account the ECE along with the F1 score and select a student trained for 2 epochs with $\lambda = 2$ and a learning rate of $2 \times 10^{-4}$. This student model reaches an F1 score and AUC of 0.982 and 0.922 and an ECE and Brier score of 0.018 and 0.024, respectively.

### 5.2 Results

Having trained a teacher and student model, we can now analyze and compare them more in-depth. To this end, we evaluate the posterior predictive distributions, predictive performance, inference speed, uncertainty quality, and calibration performance.

#### 5.2.1 Distribution Matching

We first analyze the posterior predictive distributions, i.e., the probability predictions output from each model for the test set. Figure 1 displays both the teacher and student's approximate posterior predictive distribution, together with the average predictive entropy. The distributions are visualized on the logarithmic scale since they are heavily skewed towards predictions close to 0 and 1, allowing for visually assessing their similarity. We observe a highly similar posterior predictive distribution for both models. The teacher posterior distribution features two peaks, one high and sharp and the other smaller and softer, corresponding to high densities around the extremes. This implies that the teacher model is very confident about a large share of its predictions. Although also sharply peaked, the student posterior features a slightly wider base than the teacher posterior distribution, suggesting that the student has a slightly higher level of uncertainty in its predictions. Both models have the same average entropy, indicating how well the student matches the teacher's output. Hence, we can train a student model using the uncertainty-aware distribution distillation framework that successfully captures the posterior predictive distribution of the teacher model.

#### 5.2.2 Predictive Performance

Second, we compare the predictive performance of each model, which we evaluate in terms of the F1 score and the AUC. In addition, we perform a run-time comparison, assessing the inference speed of both models. For this purpose,

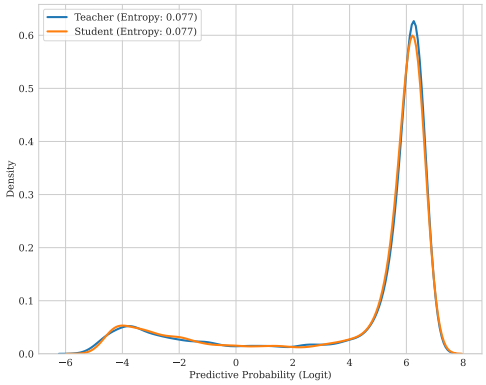



Figure 1: Posterior Predictive Distribution of Teacher and Student Model

we vary the number of MC dropout samples used for obtaining the teacher's inference results while using a consistent number of MC samples, fixed at 50, for the student model. Figure 2a shows how the teacher's F1 score varies by the number of MC dropout samples and compares it with the student's. We can see a general trend of an improved F1 score when using multiple samples, but the improvement is not consistent for further increased sample counts. For any number of MC dropout samples, the student's F1 score is above the teacher's, suggesting enhanced predictive performance, although the difference is relatively marginal. Analogously, Figure 2b visualizes this relationship for the AUC. We note a similar pattern for the increase in the teacher's performance, although the difference between a single stochastic pass and the averages of multiple samples is significantly larger. Additionally, the AUC continues to increase slightly with the number of samples. In contrast to the F1 score, the student's AUC is lower than the teacher's for any number of MC dropout samples tested greater than one.

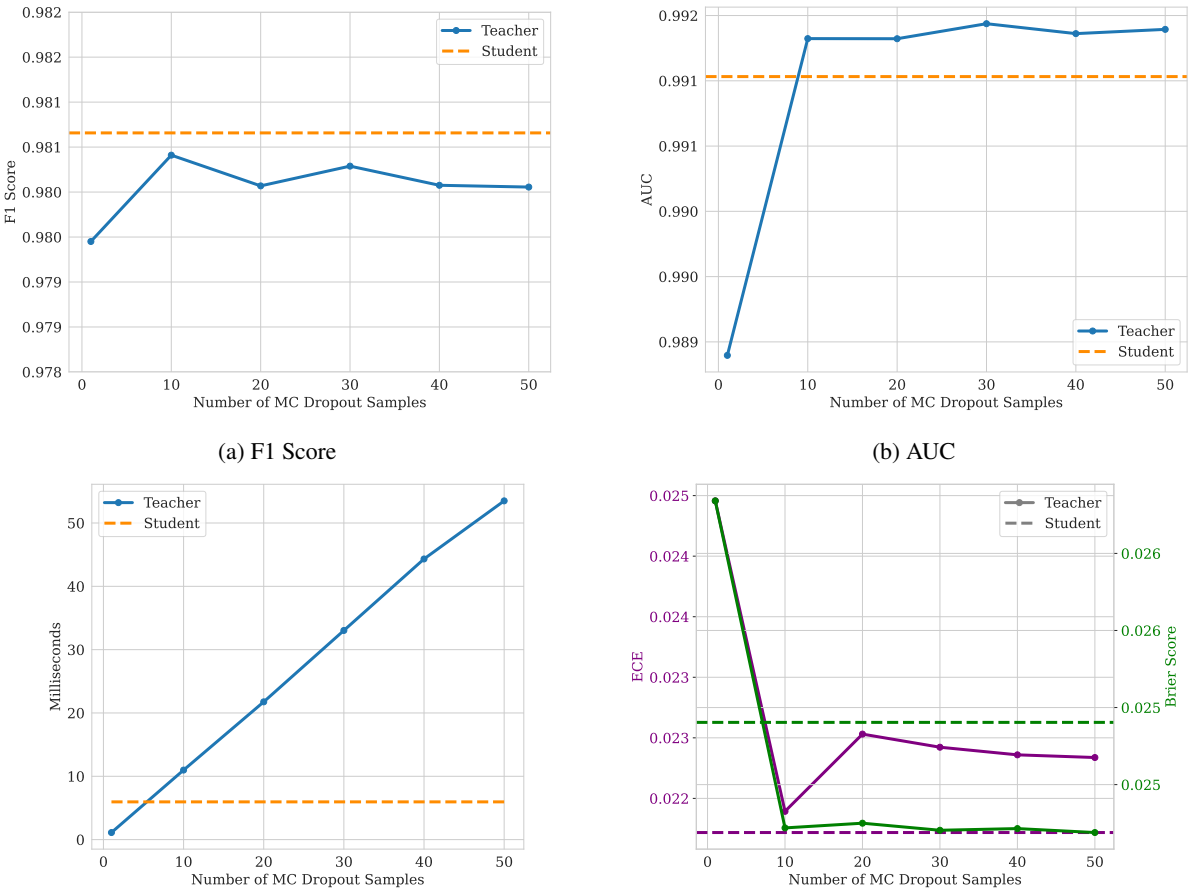
A core concern of Shen et al. (2021) is to obtain a student model that generates uncertainty estimates in real-time. Hence, we compare the average inference speed of the teacher and student model over the number of teacher MC dropout samples. This is depicted in Figure 2c. As expected, the teacher's average inference time increases almost linearly with the number of MC dropout samples. We can reduce the inference time by using fewer samples, which, however, negatively impacts the predictive quality, as noted by Shen et al. (2021) and observed above. Theoretically, the inference speed of the MC dropout teacher can be optimized by caching the network output before the first dropout layer (Shen et al., 2021). However, this is more complicated for BERT, as the architecture features several dropout layers throughout the model. Hence, we do not expect caching to increase inference speed significantly. As explained previously, the student's inference time is negatively affected by the slight overhead due to MC sampling on the logits to generate uncertainty estimates. Nevertheless, this overhead is substantially lower than that of MC dropout sampling. We thus find that the student can match the teacher in predictive performance, as measured by the F1 score and AUC. The student achieves this at a fraction of the teacher's inference time when considering the computational overhead of MC dropout inference.

### 5.2.3 Uncertainty Quality and Calibration

Finally, we investigate the uncertainty quality of the two models, which is closely related to their calibration. We compare teacher and student in terms of the ECE and Brier score and illustrate their respective calibration with a calibration curve.

Figure 2d displays the ECE and Brier score for the teacher and student model across varying numbers of teacher MC dropout samples. The ECE is very low overall, suggesting that both teacher and student are well-calibrated for the given task. The teacher's ECE decreases consistently with the number of samples, apart from an outlier, indicating an improvement in calibration and, by extension, uncertainty quality. However, the student exhibits a lower ECE than the teacher for any number of samples. Similar to the ECE, both models show excellent performance in terms of the Brier score. The observed pattern resembles a mirrored version of that for the AUC, with the teacher's Brier score decreasing slightly with the number of samples, outperforming the student when drawing more than a single MC dropout sample.

We can further investigate model calibration with the help of calibration curves. By visualizing the relationship between predicted probabilities and actual outcomes, they can reveal whether a model tends to under-predict or over-predict actual outcomes in different ranges of predicted probabilities. For a perfectly calibrated model, the predicted probabilities should match the actual frequencies exactly. For example, if it predicts an event with a 70 percent probability, it should occur about 70 percent of the time.




(a) F1 Score

(b) AUC

(c) Inference Speed

(d) ECE and Brier Score

Figure 2: Performance of Teacher and Student Model

We compare the calibration curves of the teacher and student model in Figure 3. The teacher model is well-calibrated in the lower probabilities but tends to slightly over-predict the actual outcomes, meaning that events occur less frequently than predicted. For higher probabilities, the teacher significantly over-predicts the actual outcomes, while the teacher's calibration is nearly perfect for high-confidence events. The student model's calibration curve generally shows a less pronounced deviation from perfect calibration, indicating more reliable predicted probabilities than the teacher. While the deviation can be higher for some probability ranges, it is more consistent than the teacher's. The student model tends to be well-calibrated in the lower probabilities but slightly over-predicts the actual outcomes. Generally, the student tends to over-predict, but less so than the teacher. Hence, the student is better calibrated for most probabilities but exhibits slightly worse calibration than the teacher at the extremes.

Good local calibration around the 0.5 threshold implies that a model can reliably distinguish between the positive and negative classes. This is crucial in practice as it directly affects the decision-making based on the model predictions. A model that is well-calibrated in this region might be preferable even if it is less calibrated elsewhere (Niculescu-Mizil and Caruana, 2005). Nevertheless, the overall calibration of a model is a global property that considers the entire probability range.

We conclude that the student exhibits slightly better uncertainty quality and calibration than the teacher, keeping in mind practical reliability requirements.

### 5.3 Summary

We successfully applied the uncertainty-aware distribution distillation framework to the BERT architecture for a text classification problem. To this end, we optimized a Bayesian teacher, employing MC dropout to obtain uncertainty estimates, generated a transfer set incorporating the teacher's aleatoric and epistemic uncertainty, and distilled a

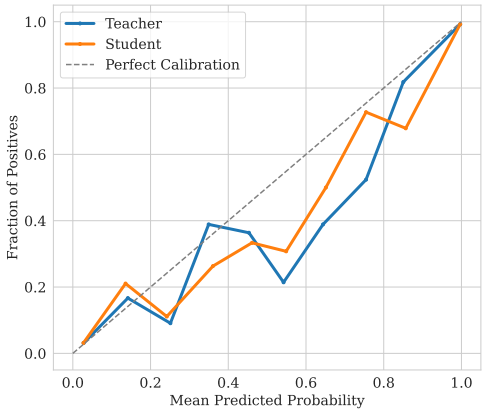



Figure 3: Calibration of Teacher and Student Model

deterministic student to incorporate these uncertainties. Addressing Research Question 1a, we found that the distilled BERT student successfully learned the approximate posterior predictive distribution of the MC dropout BERT teacher. Specifically, the training process enabled the student model to replicate the teacher's outputs of aleatoric and epistemic uncertainty effectively. Consequently, the student model could closely match the teacher's predictive posterior distributional shape. To examine Research Question 1b, we compared the teacher's and student's predictive performance using metrics such as the F1 score and AUC. The results indicated that the student model matched the teacher's performance. Additionally, the student performed inference significantly faster and could generate results in near real-time. Both the teacher and student models demonstrated similar levels of uncertainty quality, as found when exploring Research Question 1c. Notably, the student model exhibited slightly better calibration, as evidenced by the calibration plot and a lower ECE. Comparing the predictive distributions, we observed that the student's predicted probabilities tended to be softer and thus more uncertain, reflected by the student's slightly elevated Brier score. This can be interpreted as the student being more "careful" with its predictions, enhancing its overall uncertainty quality and thus outperforming the teacher. Throughout, our results align with those of Shen et al. (2021) regarding predictive performance, runtime, and uncertainty quality.

## 6 Robustness Study

Real-world applications of deep learning models often expose them to distributional shifts of the test data (Ovadia et al., 2019). This requires models to be robust to changes in the data distributions and, more importantly, reliable uncertainty estimates to determine whether a model's predictions can be trusted. This section examines the effect of distributional shifts on the predictive performance and uncertainty quality of the teacher and student models obtained above. It thus aims to answer Research Question 2. For this purpose, we employ a robustness study. Investigating the models' robustness helps verify the results of Section 5 and further assesses how they compare in quantifying uncertainty. This has additional direct implications for applying the uncertainty-aware distribution distillation framework in a practical hate speech classification context.

### 6.1 Methodology

We propose a methodology suited explicitly to our problem setting. It is heavily based on existing approaches to evaluate the robustness of models for CV tasks such as that of Ovadia et al. (2019). Specifically, we aim to create a structured approach for inducing a distributional shift by varying the noise level of text data. Consequently, we want to measure how the teacher and student's performance and uncertainty estimates change with respect to the type of noise and noise level.

Practical and especially safety-critical applications require calibrated predictive uncertainty. Distributional shifts are a common issue in real-world applications (Quinonero-Candela et al., 2022). As the shift from the original training data grows, it becomes increasingly critical to understand questions of risk, uncertainty, and trust in a model's output (Ovadia et al., 2019). A robustness study helps enable a precise risk assessment and provides a well-established and methodologically sound approach to answering our research question.

We can categorize dataset shifts into distinct groups: We speak of a covariate shift when the feature distribution $p(\boldsymbol{x})$ changes, while the conditional distribution of the label $p(\boldsymbol{y}|\boldsymbol{x})$ remains fixed (Izmailov et al., 2021a). On the other hand, a label shift occurs when the label distribution $p(\boldsymbol{y})$ changes while $p(\boldsymbol{x}|\boldsymbol{y})$ is fixed, which corresponds to annotation noise (Yu et al., 2020). Finally, in open set recognition and subpopulation shift, new classes may appear at test time (Scheirer et al., 2013) or the frequencies of data subpopulations change (Koh et al., 2021). In our study, we consider the case of a covariate shift where only the test data is corrupted by some noise; the test data originate from the same distribution as the training data but are subsequently corrupted by some transformation (Izmailov et al., 2021a). Here, we desire the model predictions to become more uncertain with a higher shift (Ovadia et al., 2019).

Since no standard framework exists for systematically evaluating model robustness in a social media text classification context, we propose an original approach suited to our specific problem setting. We base this approach on Ovadia et al.'s (2019) methodology and adapt it to the NLP domain and specifically to the hate speech detection dataset employed in our study. Robustness in the existing literature typically refers to ways to enhance model robustness by modifying the training process by perturbing the training data or exposing a model to adversarial examples during training (Wang and Culotta, 2021). Prior research additionally uses adversarial examples to measure the robustness of already trained models (Jin et al., 2020; Liang et al., 2018). Although established as an effective way of measuring model robustness in the CV domain, systematic robustness studies, as proposed here, are rare in the NLP domain. In short, our approach allows for systematically introducing and increasing the noise in text data. We evaluate the teacher and student on perturbed variations of the test dataset and measure how model performance, calibration, and uncertainty quality change with respect to noise type and level.

As we are interested in how the teacher and student model compare when evaluated on data affected by a covariate shift, we introduce noise exclusively into the test dataset. Again, we do not aim to improve model robustness through training on augmented data but to assess the robustness of the teacher and student. Hence, as done previously, we require the models to be trained and optimized on unmodified training and validation data.

Noise is frequently used as an equivalent of augmentation. We differentiate between document-level, sentence-level, word-level, and character-level noise, which can be natural or synthetic (Belinkov and Bisk, 2018). Natural word-level noise includes synonym replacement or the introduction of typos and misspellings (Wei and Zou, 2019; Belinkov and Bisk, 2018). In contrast, examples of synthetic word-level noise are random word insertion, random word swap, and random word deletion (Wei and Zou, 2019). Examples of synthetic character-level noise include random character swap, word character randomization, and keyboard typo replacement (Belinkov and Bisk, 2018). In our robustness study, we employ exclusively word-level noise and a combination of natural and synthetic noise types. As for natural word-level noise, we decide to use part-of-speech guided word replacement (Tang et al., 2019), where we randomly replace a word with another of the same part-of-speech (POS) tag with probability $p$, and synonym replacement (Wei and Zou, 2019), where we randomly replace a word with one of its synonyms, again with probability $p$. We include three types of synthetic noise, following the definitions of Wei and Zou (2019), namely random deletion, random swap, and random insertion, where we randomly delete, swap, or insert words within a sequence, all with probability $p$. We vary the noise level by changing $p$, where we consider a range from 0.05 to 0.5 in steps of 0.05.

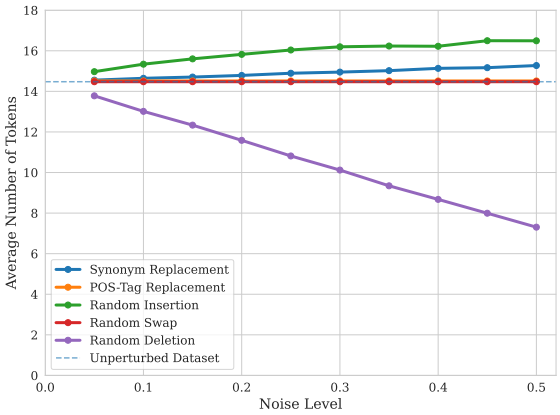



Figure 4: Effect of Noise on Sequence Length

Figure 4 illustrates the effect of applying the different perturbations to the test dataset. The average sequence length, measured in the average number of tokens, is a proxy measure of the distributional shift, which is not directly quantifiable. The figure plots the average sequence length against the noise intensity $p$, separated by noise type, and

**Original**
sighs of relief from beijing guoan fans <hashtag> chinas govt as club nips japanese rival <number> <number> in tense match <hashtag> football <url>

---

**Synonym Replacement**
~~sighs~~suspire of ~~relief~~allevation ~~beijing~~peiping guoan ~~fans~~devotee <hashtag> ~~chinas~~mainland china govt as ~~club~~society nips japanese rival <number> <number> in tense ~~match~~peer <hashtag> football <url>

**POS-Tag Replacement**
sighs of relief from beijing guoan ~~fans~~b*tch <hashtag> chinas govt as ~~club~~oh ~~nips~~o japanese rival <number> <number> in tense ~~match~~prob <hashtag> football <url>

**Random Insertion**
sighs easement mainland china of amp relief from tense up twinge beijing guoan fans <hashtag> chinas govt as club nips japanese rival <number> <number> in tense match <hashtag> football <url>

**Random Swap**
sighs <hashtag> nips match beijing of fans club chinas govt from guoan relief <hashtag> rival <number> <number> in tense as japanese football <url>

**Random Deletion**
~~sighs~~ of relief ~~from~~ beijing guoan fans <hashtag> ~~chinas~~ govt ~~as~~ club ~~nips~~ ~~japanese~~ rival <number> <number> ~~in~~ tense match <hashtag> ~~football~~ <url>

Figure 5: Illustration of Noise Effect on Preprocessed Input Text

for the unperturbed test dataset. Random deletion noise results in the strongest distribution shift. As expected, the sequences become shorter as words become more likely to be removed. The average sequence length does not change with the noise level for random swap noise and POS-tag-guided replacement. Neither noise type adds or removes words from a sequence, as they swap the words within a sequence or replace them with other singular words, respectively. Synonym replacement noise leads to a slight increase in the average number of tokens, as it operates on singular words, and synonyms may consist of more than one word. We observe the most substantial increase in average sequence length for random insertion noise, which aligns with our expectations as it does not delete or replace but only adds words to a sequence. To illustrate the effect of the different noise types, we showcase the results of applying each to the same sequence at the highest noise intensity of $p = 0.5$ in Figure 5.

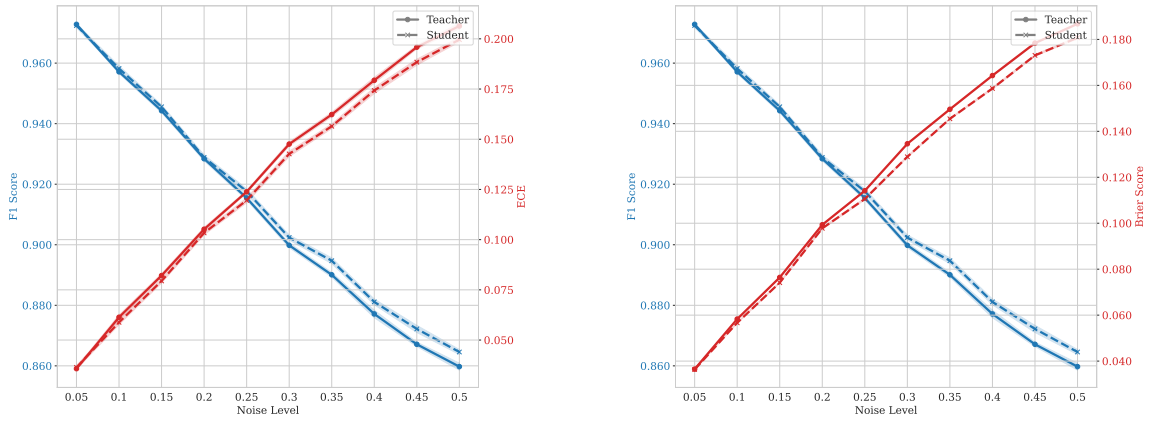
In line with Ovadia et al. (2019), we generally expect a decline in model performance when evaluated on noisy test data. Any noise type introduces distortions to the input-output signal, making accurate classification more difficult. Interpreting the shift in average sequence length as a proxy measure of distributional shift, we anticipate random deletion and random insertion noise to have the strongest negative impact on performance. We hypothesize, informed by the results of Shen et al. (2021), that the student is more robust to noise, resulting in a weaker decline in performance compared to the teacher.

Using the perturbed data, we conduct the robustness study by evaluating the fine-tuned teacher and student models on each type-level variant of the test data and recording the results. We obtain inference results via MC dropout and MC sampling for the teacher and student models, respectively. The predictive performance is measured using the F1 score, and we evaluate the uncertainty quality and calibration with the ECE, Brier score, and predictive entropy, as also done by Ovadia et al. (2019) and Shen et al. (2021). We repeat this process 20 times to compute each metric's mean and standard deviation to make statements concerning significant improvements or deteriorations in predictive and uncertainty performance.

### 6.2 Results

We turn first to analyzing the results for perturbations with natural noise. Figure 6 shows the effect of POS-tag replacement noise on the teacher and student model's mean F1 score, ECE, and Brier score. It additionally indicates the 95 percent confidence intervals around these mean performance values. POS-tag replacement noise strongly affects both predictive performance and uncertainty quality. Generally, the F1 score decreases significantly, while the ECE

and Brier score increase substantially with the noise level. We observe that the student significantly outperforms the teacher in terms of F1 score for higher noise levels. Similarly, the student exhibits significantly better calibration and uncertainty quality than the teacher for higher noise levels, as indicated by a lower ECE and Brier score.



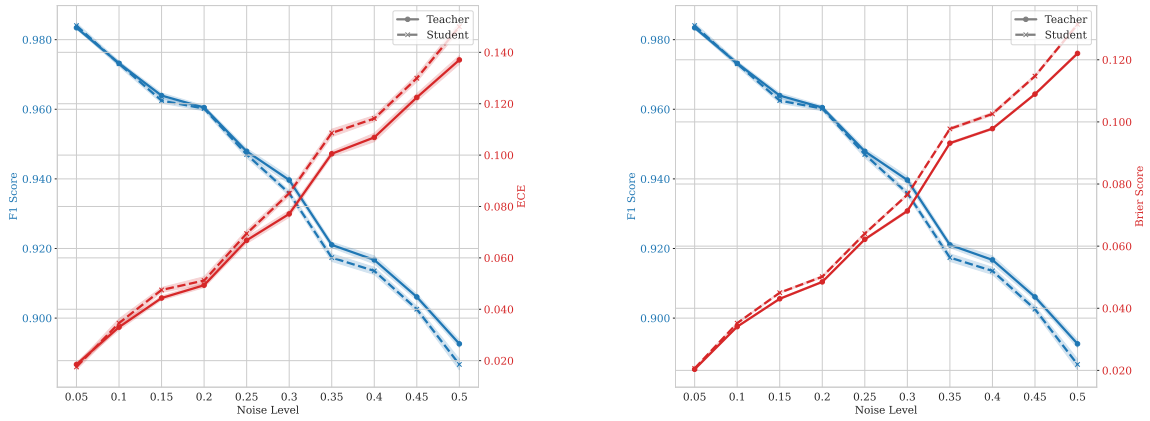

(a) F1 Score vs. ECE

(b) F1 Score vs. Brier Score

Figure 6: Effect of POS-Tag Replacement Noise

In contrast, the teacher tends to outperform the student when exposed to synonym replacement noise. This is visible in Figure 7, which shows how synonym replacement noise affects both models' performance. While the F1 score decreases as the noise level increases, the ECE and Brier score increase significantly. However, the changes are of a lesser magnitude compared to the POS-tag replacement noise. The teacher model consistently and significantly outperforms the student in predictive performance and ECE. This is also reflected in the Brier score, which shows a pattern similar to the ECE.

Next, we examine the effect of synthetic noise, starting with random insertion noise, as shown in Figure 8. We note that the predictive performance is above the baseline for all noise levels, affecting the ECE and Brier score. The F1 score decreases slightly but significantly for teacher and student, while the ECE exhibits a slight upward trend. On the other hand, we observe a marginal but significant increase in the Brier score. Overall, we find no clear pattern of one model significantly outperforming the other in predictive performance, uncertainty quality, and calibration.



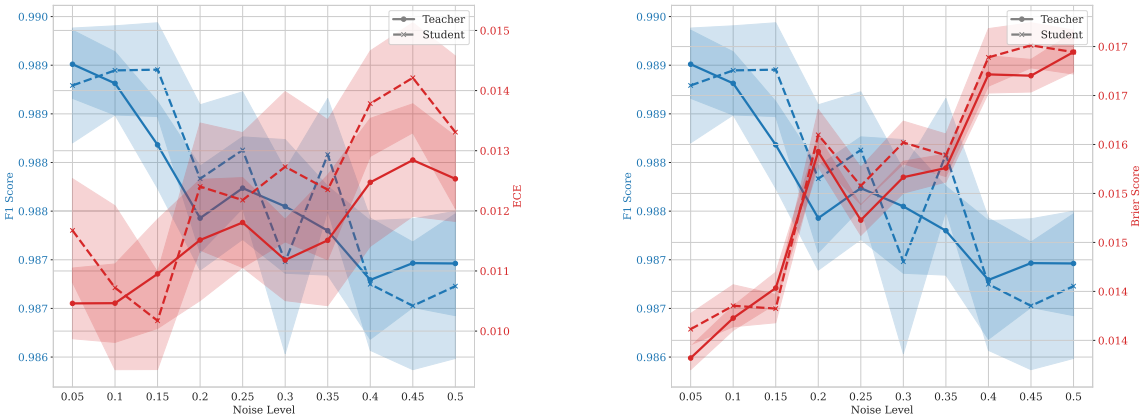

(a) F1 Score vs. ECE

(b) F1 Score vs. Brier Score

Figure 7: Effect of Synonym Replacement Noise

Similarly, random swap noise seemingly elevates the general performance level. We visualize this effect in Figure 9. Again, we observe a small relative decrease in the F1 score across both models, which is statistically significant. The ECE does not change significantly with the noise level, whereas the Brier score increases only slightly but significantly. As for random insertion noise, no model outperforms the other for any metric.

Finally, we analyze the effect of random deletion noise on predictive and uncertainty performance, shown in Figure 10. As previously observed for POS-tag guided and synonym replacement noise, the F1 score decreases substantially and significantly, while the ECE and Brier score increase greatly. Notably, these changes are the strongest observed across all noise types. The student significantly outperforms the teacher regarding predictive performance, and the performance gap increases with the noise level. We note a similar pattern for the ECE and Brier score.

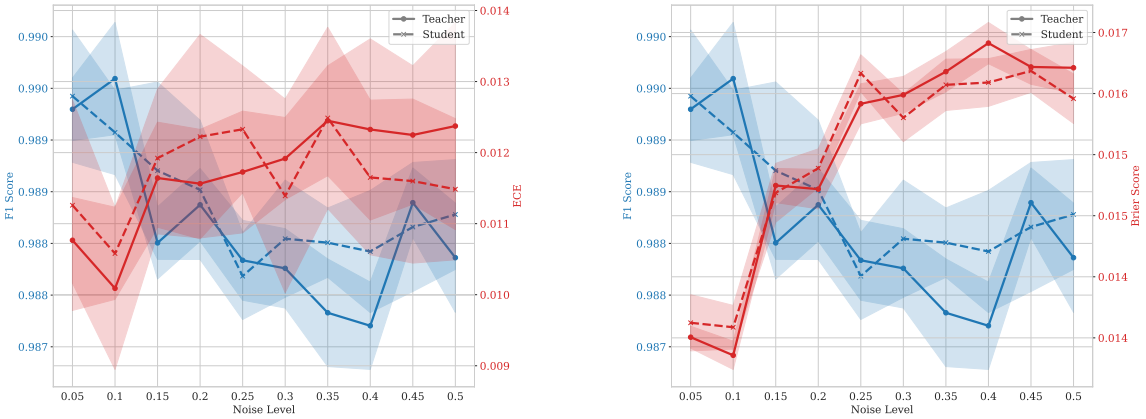



(a) F1 Score vs. ECE

(b) F1 Score vs. Brier Score

Figure 8: Effect of Random Insertion Noise

In summary, POS-tag replacement, synonym replacement, and random deletion noise strongly negatively affect predictive performance, calibration, and uncertainty quality, consistent across teacher and student models. Surprisingly, random insertion and random swap noise positively impact model performance, but the pattern is unclear overall. We observe a general trend of the student outperforming the teacher in predictive performance, as measured by the F1 score, and uncertainty quality and calibration, as measured by the ECE and Brier score. This trend not only suggests that the student's predictions are, on average, equally or slightly more reliable than the teacher's in the face of covariate shifts but also reaffirms the success of the distillation process. These results further show that the student model possesses enhanced robustness to noise contained in the data relative to the teacher model. However, we do not observe a consistent pattern across all noise types. While for POS-tag replacement and random deletion noise, the student model outperforms the teacher model in predictive performance and uncertainty quality, the student model tends to have worse performance and uncertainty quality for synonym replacement. There appears to be no clear pattern for random swap and insertion noise. Overall, the random deletion of tokens and their POS-tag-based replacement are by far the strongest types of noise, followed by synonym replacement.



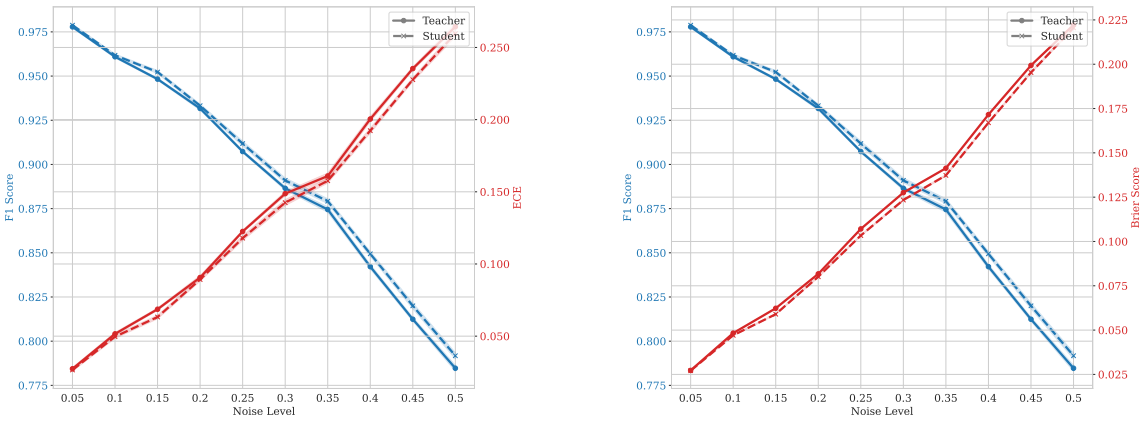

(a) F1 Score vs. ECE

(b) F1 Score vs. Brier Score

Figure 9: Effect of Random Swap Noise

As noise increases, both the ECE and Brier score tend to increase, indicating a significant deterioration in uncertainty quality, calibration, and predictive performance. However, the two metrics paint a slightly different picture. The ECE measures the agreement between predicted probabilities and observed outcomes, so as noise increases, predictions naturally get worse overall, and the ECE increases. On the other hand, the Brier score penalizes deviations from the true outcomes. Hence, like for the ECE, the Brier score increases due to worse predictions. Therefore, we expected both to increase with the noise level as a model's predictions become less accurate.

However, we notice a higher relative increase in the Brier score than in the ECE for random insertion and random swap noise. Investigating this phenomenon requires an in-depth understanding of the two metrics, specifically the Brier score. The Brier score can be decomposed into uncertainty, reliability, and resolution (Murphy, 1973). We can separate the components into calibration, which corresponds to reliability, and refinement, the combination of resolution and uncertainty. We suspect that the Brier score is affected by the sharpness or peakedness of the predictive distribution via the refinement component. As the noise increases, the predictive sharpness decreases as the models become more uncertain. Even if calibration remains constant, the Brier score increases as a result.




(a) F1 Score vs. ECE

(b) F1 Score vs. Brier Score

Figure 10: Effect of Random Deletion Noise

We verify this theoretical explanation by analyzing the distribution of the predicted probabilities with increasing noise levels. Here, we can measure the sharpness of the predictive distribution using the average predictive entropy. The higher the average entropy, the less peaked the predictive distribution and, thus, the more uncertain the model, which is what we expect for increasing noise. The lower the entropy, the more peaked and the more confident the model is. An increase in entropy for the noise types in question would provide us with an explanation for the observed results.

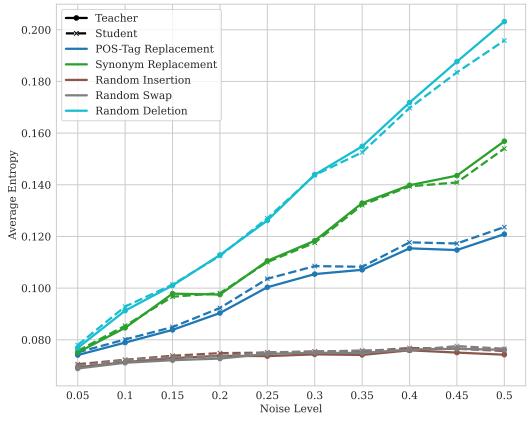



Figure 11: Effect of Noise on Predictive Entropy

Figure 11 shows how the average predictive entropy of the teacher and student evolves with increasing noise for the different types of noise. We observe an increase in the average entropy as the noise intensifies, but this increase differs

between noise types. In addition, we notice a difference in the average entropy between teacher and student, but the pattern is inconsistent. A softer predictive distribution due to increasing noise correlates with higher ECE and Brier score, which aligns with our expectations. If a model was previously making very confident correct predictions, e.g., predicting a high or very high probability for the positive class, which corresponds to the true label, then making these predictions softer, in effect making the predicted probabilities less extreme, leads to an increase in the squared predictive error and results in a higher Brier score. This aligns with our observation that the largest increase in the Brier score for random deletion noise coincides with the most significant increase in entropy. Similarly, noise renders the classification task more difficult, leading the model to make predictions that are, on average, less reliable. This mismatch between confidence and accuracy is what the ECE measures. Hence, an increase in entropy due to noise is directly related to a rise in ECE, as both reflect growing uncertainty in the model's predictions. Notable exceptions are the random insertion and random swap noise, for which the predictive entropy does not significantly increase with the noise level. This coincides with the previously observed marginal changes in the ECE and the Brier score. However, this analysis does not allow us to explain the observed difference in trends between the two metrics for these noise types.

In summary, perturbing the data degrades their quality, making it more difficult for the models to predict the correct class accurately. Higher noise results in softer output probabilities and, thus, higher average predictive entropy as the models become more uncertain, spreading out their predictions rather than concentrating them. Generally, with an increasing noise level, the predictive performance, as measured by the F1 score, deteriorates, predictive refinement worsens, as indicated by an increasing Brier score, and the models become more uncertain and less calibrated, as illustrated by a higher ECE.

### 6.3 Student Augmentation

While the previous results indicate that we can distill a student model using uncertainty-aware distribution distillation that retains the teacher's predictive performance and achieves better calibration and noise robustness, Shen et al. (2021) find that training the student on a transfer dataset that differs from the data the teacher was initially trained on can further improve the student's uncertainty quality. However, they apply their framework to image datasets to which they natively apply augmentations for training the teacher model, as is typical in the CV domain. To achieve a difference in the datasets used for training teacher and student without having to acquire a different labeled dataset, they hold out some augmentations exclusively for training the student model. We did not incorporate augmentations into the training process of the teacher model, as their use is less common in the NLP domain, and we wanted to comply with a setup closely resembling a practical setting. Therefore, we infer suitable augmentations for training the student model from the robustness study.

The results of our robustness analysis suggest that the student model might benefit from augmenting the transfer training data with several of the employed noise types. In particular, the student struggles the most when exposed to data heavily affected by synonym replacement noise. On the other hand, random insertion and random swap noise seemingly facilitate the correct classification of sequences, as they elevate the F1 score above the baseline level. Specifically, the random swapping of words likely decreases the reliance on grammatical structures. As hate speech typically hinges on single words (Davidson et al., 2017), this augmentation lends itself particularly well to the text classification task at hand. Hence, we propose lightly augmenting the transfer dataset with random swap noise. An augmented transfer dataset can be created by applying the selected augmentation to the teacher samples obtained as described in Section 5.1.2. We will then fine-tune the student model on this augmented dataset and evaluate its performance on the test set. We expect this new student to exhibit better generalization performance and calibration, as it will have been trained on a more diverse dataset, leading to a more robust model overall.

#### 6.3.1 Methodology

As outlined above, obtaining an augmented student involves applying transformations to the transfer dataset and fine-tuning a student on this augmented dataset. We subsequently evaluate the resulting student model on the unmodified test set.

We augment the transfer dataset using a relatively low level of the single selected noise type. The impact of noise, as used in this work, is significantly stronger compared to augmentations typically used in the CV domain. Since it is applied on the individual word level, nearly every sequence is affected, even for low noise levels. Hence, we restrict the augmentation to a single noise type and a low noise level. Specifically, we apply random swap noise at a level of $p = 0.1$.

As the underlying training data has changed, we again optimize the student hyperparameters using a grid search. As done for the initial student, we optimize the loss weight $\lambda$, the learning rate, and the number of training epochs, for which we use the same value ranges as before. We fine-tune the student for each possible hyperparameter combination

and then evaluate it on the test set to determine the best configuration. Here, we find a model with a loss weight $\lambda = 2$, a learning rate of $2e{-}4$, and trained for 2 epochs to perform best in terms of the F1 score. This optimal configuration differs only in the number of epochs compared to the original student. The resulting augmented student achieves a performance on the test dataset of an F1 score and AUC of 0.982 and 0.992 and an ECE and Brier score of 0.018 and 0.024, respectively.

### 6.3.2 Results

In the following, we perform a comparative analysis of the augmented and the original student regarding predictive performance, uncertainty quality, and calibration. We expect the augmented to outperform the original student in terms of all metrics, in line with the findings of Shen et al. (2021).

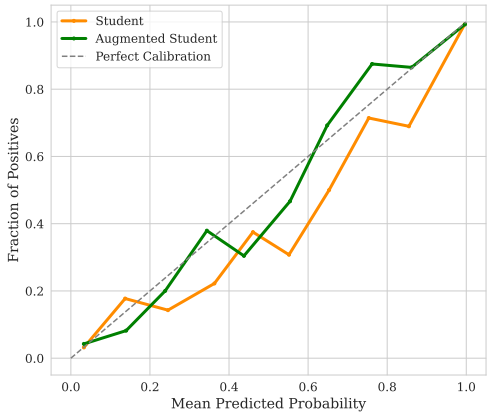



Figure 12: Calibration of Student and Augmented Student Model

The augmented student indeed outperforms the original student in the F1 score, ECE, and Brier score while achieving equal performance in terms of AUC. This suggests that the augmentation yields stable or slightly improved predictive performance while enhancing model calibration and uncertainty quality. Similarly, the augmented student significantly outperforms the teacher in all relevant metrics. The augmented student's low ECE is indicative of high uncertainty quality. Compared to the original student, it has a higher average predictive entropy, attesting to a softer posterior predictive distribution and generally higher uncertainty. We can further investigate the calibration by comparing the calibration curves of the augmented and original student, which we do in Figure 12. The augmented student almost consistently exhibits superior calibration and is nearly perfectly calibrated for low and very high probabilities. Further analysis of the differences in predicted probabilities between the two students reveals that the augmented student tends to predict extreme probabilities, close to 0 and 1, less frequently than the original student. Generally, it more frequently predicts lower probabilities, leading to the slight observed over-prediction in the 0.35 to 0.6 probability range in Figure 12 and the under-prediction of higher probabilities in the 0.6 to 0.8 range.

Hence, augmenting the transfer dataset enhances the student's predictive performance and uncertainty quality. Augmentations efficiently improve the results of uncertainty-aware distribution distillation in this text classification context.

## 6.4 Summary

In studying the models' robustness to covariate shifts, we found that noise negatively impacted the predictive performance. With regard to Research Question 2, the magnitude of this impact differed substantially between different types of natural and synthetic noise. In addition, the reliability and uncertainty of the models deteriorated as the noise level increased. In general, the student performed better in the face of noise, both in predictive performance and uncertainty quality. This led us to conclude that the student was slightly more robust to noise than the teacher. This was likely due to the student making more careful predictions and being better calibrated as a result, as we observed throughout this and the previous section. The findings of our robustness study aligned with those of prior research. Similar to Ovadia et al. (2019), we observed a degradation in model accuracy as the data became increasingly shifted, which coincided with an increase in predictive entropy. Furthermore, the originally well-calibrated student managed to, on average, outperform the teacher in terms of model calibration. These results thus serve as a strong indicator of the superiority of the student model and highlight the benefits of the uncertainty-aware distribution distillation framework.

In the following excursion, we applied and verified an approach to improve the uncertainty-aware distribution distillation method suggested by Shen et al. (2021). For this purpose, we lightly augmented the transfer dataset with noise derived from the robustness study. We performed a grid search to determine the optimal hyperparameter settings in which we fine-tuned a new student model on the modified transfer dataset. Subsequently, we compared the resulting model to the original student obtained in Section 5.1.3. Here, we found that training a student on the augmented transfer dataset improved predictive and uncertainty performance. These results suggested that augmenting the transfer dataset efficiently enhances the distillation framework, which is highly relevant for any applications in which we desire optimal performance. In addition, we were able to confirm the findings of Shen et al. (2021). Nonetheless, this approach requires access to more labeled data, which can be costly in many practical settings, or using context-appropriate augmentations.

# 7 Out-of-Distribution Analysis

Practical data is often non-stationary, meaning the underlying feature distribution changes over time (Gama et al., 2014). Once deployed, machine learning models are additionally frequently evaluated on data that differ in feature and target distribution and even on data with an entirely different target than what the model has been trained on. In both instances, it is crucial for a model to be able to express its uncertainty, which is the focus of the following section.

To address Research Question 3, we investigate how the teacher, the original student, and the augmented student perform in terms of uncertainty when evaluated on data whose distribution differs significantly from that of the training dataset in an OOD analysis. To this end, we introduce two novel datasets that are contextually related to the training dataset but are recorded in a different time period and for a different purpose, respectively.

This OOD analysis extends the preceding robustness study and provides deeper insight into the models' uncertainty quality. Detecting OOD inputs is highly relevant for practical applications and especially important for hate speech detection settings where model reliability is essential.

## 7.1 Data

We want to investigate how the different models perform when seeing OOD data. Each model is trained on in-distribution (ID) data, the original training dataset, and evaluated on OOD data. In other domains, such as the CV domain, the concept of OOD data is relatively straightforward: Considering the traffic scene dataset used by Shen et al. (2021), OOD can refer to held-out classes (e.g., cyclists) or scenes from a different location (e.g., a different city). This distinction is more complex in the NLP domain, in part because it is difficult to measure the difference between different types of text data. Here, OOD can refer to different types of documents (e.g., books vs. articles) or, given a document type, to different domains (e.g., product reviews vs. social media posts) (Van Landeghem et al., 2022). Additionally, temporal differences exist, even within a specific document type-domain category; as any language changes over time, the "distribution" of the same type of text data constantly shifts. Finally, the language itself could differ in its distribution, e.g., through a dialect or being a different language altogether within a specific range, e.g., German and Swiss German or German and Dutch.

We consider several related studies in the existing literature to select the OOD datasets. In a similar analysis, Shen et al. (2021) use a disparate dataset with classes overlapping with the original training set. Meanwhile, Ovadia et al. (2019) additionally consider an OOD dataset with labels different from the training data. Van Landeghem et al. (2022) focus more on practical domain similarity, arguing that a setting where in-domain and out-of-domain datasets are at least slightly related is more realistic than measuring out-of-domain detection in totally disparate domains.

Since the primary focus of this analysis is not measuring predictive performance, we do not require the OOD datasets to feature the same classes as our training dataset. To ensure contextual relatedness, we use different datasets consisting of Twitter/X posts.

Text classification models deployed in practice must work consistently well, even in the face of emerging vocabulary trends. In this setting, the underlying type of data and the problem setting remain the same, but the data distribution might change significantly. Hence, we consider an OOD dataset with a contextually similar target and a closely related feature distribution. It differs considerably from the training data because they were collected during a different time frame. Specifically, we use English tweets, collected primarily in 2018 by Basile et al. (2019), with a focus on hate speech against immigrants and women. Each tweet is labeled either hate speech or non-hate speech. For this analysis, we use their training set, which contains around 9,000 tweets.

In practical applications, text classification models may encounter datapoints unrelated to their training data, particularly in scenarios where they are used to identify relatively infrequent events, such as hate speech comments on social media platforms. Here, the evaluation data again has the same underlying type but a contextually different target and

feature distribution. This is thus very similar to the setting described by Ovadia et al. (2019). Given that the underlying feature space of the ID training and the OOD evaluation data is closely related, this approximately corresponds to the out-of-domain analysis setting (Van Landeghem et al., 2022). We use another dataset containing English tweets collected between 2014 and 2015 by Van Hee et al. (2018). The tweets are labeled as either ironic or non-irony. We again use the specified training set for this analysis, comprising about 3,800 tweets.

### 7.2 Methodology

Our methodology is very similar to that of the robustness study. The teacher and students remain unmodified and are the same as in the previous stages. This implies that neither the teacher nor the students see any OOD data during training or fine-tuning. We evaluate all models on the two considered datasets and perform 20 runs over which we average to obtain the final results. Notably, we keep all preprocessing settings, such as the maximum token length, fixed at their fine-tuning values. This helps assess how each model performs on new, unseen data and prevents inadvertently influencing the model performance.

Concerning the evaluation of the performance of each model in the context of this analysis, we note that the predictive performance itself is less relevant. Instead, we are mainly interested in examining how different they behave in terms of uncertainty and thus compare all models to make statements regarding their relative uncertainty performance. Related studies frequently employ metrics that do not rely on ground truth labels to be flexible regarding selecting OOD datasets. We can consider these in our performance evaluation since we have access to binary ground truth labels. We measure model calibration and uncertainty quality via the ECE and Brier score. Furthermore, to better compare the uncertainty, we use the predictive entropy, as done by Ovadia et al. (2019), and the BALD score, employed for this purpose by Shen et al. (2021).

### 7.3 Results

As highlighted previously, we consider two different scenarios. The first corresponds to an OOD evaluation dataset for another hate speech detection task but with a different feature distribution. This is equivalent to evaluating the models on a data subset that was not collected together with the ID datasets and, hence, stems from a different distribution. Generally, we expect the original student to be more uncertain than the teacher, as we have shown above that it is better able to capture uncertainty. The augmented student should perform on par with or better than the original student, as indicated by a higher predictive uncertainty when evaluated on OOD inputs.

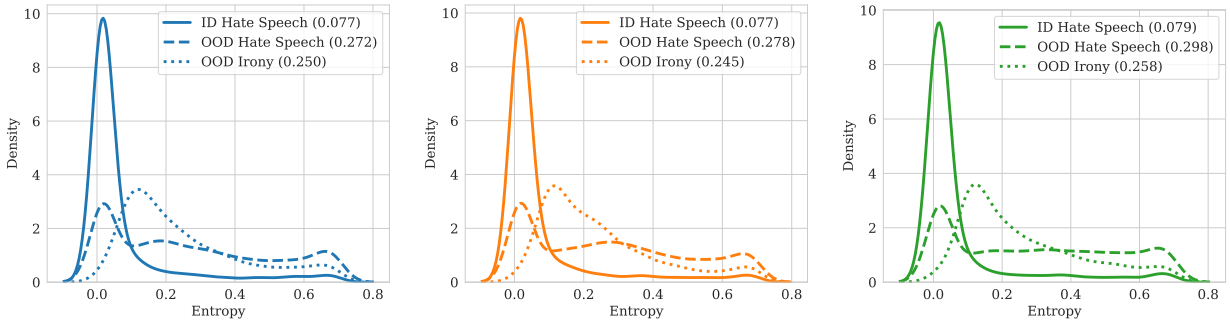
In the second setting, we consider a dataset related to the ID training data via its underlying characteristics but collected for a distinct task and, therefore, different in target and feature distribution. As before, we expect the students to be more uncertain than the teacher, with the augmented student potentially outperforming the original student. Additionally, based on Van Landeghem et al. (2022), we expect the overall uncertainty level to be higher for this than for the first setting, as this OOD dataset is even more different from the original training dataset, resulting in a higher OOD generalization gap.

| | | Dataset | | |
|---|---|---|---|---|
| Metric | Model | ID Hate Speech | OOD Hate Speech | OOD Irony |
| ECE ↓ | Teacher | 0.022 | 0.312 | 0.479 |
| | Student | 0.022 | 0.318 | 0.481 |
| | Augmented Student | 0.018 | 0.304 | 0.480 |
| Brier Score ↓ | Teacher | 0.024 | 0.299 | 0.438 |
| | Student | 0.025 | 0.302 | 0.441 |
| | Augmented Student | 0.024 | 0.292 | 0.440 |

Table 1: In- and Out-of-Distribution Performance

Table 1 displays the ECE and Brier score capturing calibration and uncertainty quality of each model on the ID and OOD hate speech and the OOD irony dataset. While the teacher and the original student achieve an equal ECE on the ID dataset, they are outperformed by the augmented student. We observe a general increase in the ECE the further away the OOD dataset distribution is from the ID training data, and this increase is very similar for all models. The augmented student outperforms the other models on the OOD hate speech dataset, whereas the original student underperforms significantly. All three models are close together with respect to the ECE on the OOD irony dataset. On the other hand, the Brier score is virtually the same for all models on the ID dataset. The augmented student outperforms the

other two models on the OOD hate speech dataset, which perform about equally. Again, all three models perform very similarly regarding the Brier score on the OOD irony dataset. However, we note a slight trend of the original student underperforming the teacher and the augmented student. While all three models struggle when evaluated on the OOD datasets, the augmented student overall performs best, followed by the teacher and, closely behind, the original student. This trend is especially evident for the OOD hate speech data. As expected, the models perform worse on the OOD irony dataset than the OOD hate speech data.




(a) Teacher (b) Student (c) Augmented Student

Figure 13: (Average) Predictive Entropy on In- and Out-of-Distribution Data

We can analyze each of the three models' predictive entropy per dataset to understand better the structure of their predictive confidence and overall uncertainty. To this end, Figure 13 shows the predictive entropy of the teacher, the original student, and the augmented student across the ID hate speech, the OOD hate speech, and the OOD irony datasets, together with each model's average entropy per dataset. We observe a similar pattern across all models: The average predictive entropy increases substantially from the ID to the OOD datasets, and the entropy distribution becomes significantly less sharp and more spread out. This strongly indicates that all models become more uncertain when evaluated on OOD data, and this uncertainty increases the farther the distribution of the OOD dataset is away from the training dataset's. The predictive entropy increases more for the student models than the teacher, suggesting that they are more uncertain than the model from which they were distilled. The original student's average entropy for the OOD irony data presents a notable anomaly, as it is lower than the teacher's. Moreover, the augmented student features a significantly higher entropy than the original student and is, therefore, more uncertain overall.

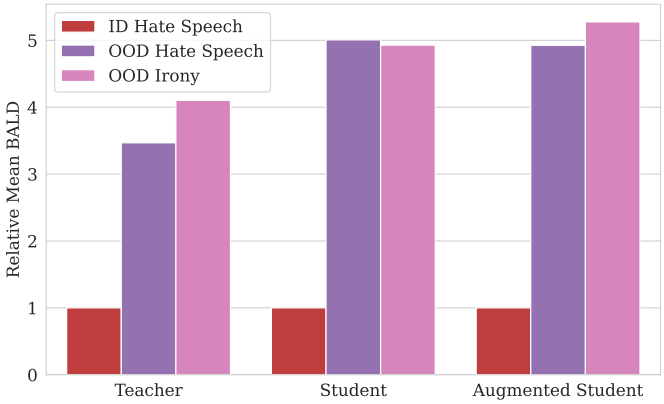



Figure 14: Relative Mean BALD on In- and Out-of-Distribution Data

Similar to the predictive entropy, the BALD score is an additional measure of predictive uncertainty. We use the relative mean BALD to quantify how much more uncertain a model is on one dataset than another. This metric is computed by dividing the mean BALD of a given dataset by that of the reference dataset. In the context of this analysis, we compute it as the mean BALD of an OOD dataset divided by the mean BALD of the ID dataset. The resulting values are displayed in Figure 14 for all models and datasets. We observe an elevated relative mean BALD on the OOD datasets, which tends to be the highest for the OOD irony dataset. It is substantially higher for the students compared to the

teacher. We prefer a model with a higher relative mean BALD for OOD data as it indicates higher predictive uncertainty. Hence, the student models significantly outperform the teacher in terms of uncertainty, as measured by this metric. Although the ranking is inconsistent, we tend to prefer the augmented student over the original student, on average. These results again indicate that the students are superior in uncertainty and uncertainty quality compared to the teacher model. As the BALD score takes into account the predictive entropy, this result was expected.

In general, all models are more uncertain the farther away the evaluation data's distribution is from that of the training distribution. This is reflected by a higher uncertainty level for the OOD irony data compared to the OOD hate speech data, which is consistent across all models. The students tend to outperform the teacher concerning uncertainty, although this trend is not always clear for the original student. The augmented student exhibits, by far, the best uncertainty performance and is preferred in almost all metrics.

### 7.4 Summary

In this section, we conducted an OOD analysis where we found that the uncertainty of all models increased substantially when evaluated on OOD datasets. Regarding Research Question 3, we observed that the models varied significantly in predictive uncertainty. Interestingly, the original student did not consistently outperform the teacher model with respect to uncertainty. The augmented student exhibited significantly higher predictive uncertainty across both OOD datasets and is thus preferred over the other models due to its enhanced OOD detection ability. Notably, higher uncertainty was correlated with a decrease in predictive accuracy, as reflected by an increase in ECE.

The performance of the original student aligned with the observations of Ovadia et al. (2019), who noted that better ID data calibration and accuracy do not typically imply better calibration on OOD data. The observed higher predictive entropy and relative mean BALD for the student models compared to the teacher is in line with the hypothesis of Shen et al. (2021) that the student learns to output less confident and more generalizable predictions by seeing the teacher's soft label distribution during training. In summary, the models differed in predictive uncertainty in favor of the augmented student model, whose uncertainty was consistently higher than its peers', leading us to conclude that it is preferred in situations with strong distribution shifts.

Our results show that uncertainty-aware distribution distillation greatly mitigates the problem of pre-trained transformers significantly underperforming in detecting OOD inputs (Van Landeghem et al., 2022). This allows for taking advantage of the predictive power of pre-trained transformer models while succeeding in correctly estimating their uncertainty on OOD inputs in practical applications.

## 8 Discussion

This study was designed to determine whether the uncertainty-aware distribution distillation framework can be applied to a transformer-based model in the NLP domain. Specifically, we investigated its applicability to the BERT architecture for a hate speech detection task.

Throughout this study, we examined the performance of the models obtained from the uncertainty-aware distribution distillation framework to determine its effectiveness when applied in this setting. We further analyzed how well the models maintain performance under covariate shifts of the test data. This involved a robustness study, in which we evaluated key aspects such as the model's predictive performance, uncertainty estimation, and calibration in scenarios where the test data were perturbed with noise. In addition, we explored creating an augmented student model by enhancing the distillation process to yield improved predictive and uncertainty performance. Finally, we conducted an OOD analysis to assess the models' capability to detect and handle inputs that significantly deviate from the training distribution. This provided insights into the effectiveness of the distillation framework in identifying and processing OOD examples, which is crucial for maintaining the models' reliability in real-world applications where unexpected inputs are common.

With respect to the first research question, we hypothesized that, using the uncertainty-aware distribution distillation framework, we would obtain a student model that can match the predictive performance and uncertainty quality of the teacher model. Indeed, we find that the student learns the posterior predictive distribution of the teacher and matches the teacher in predictive performance, uncertainty quality, and calibration. Notably, the student can generate uncertainty estimates in real-time, outperforming the teacher relying on MC dropout sampling to quantify uncertainty, which results in a significant computational overhead.

Similarly, we expected the student to be more robust to noise than the teacher as it is more careful with its predictions. Regarding our second research question, the results of this study show that noise affects the teacher and student model differently. The effect is inconsistent across noise types, but reliability and uncertainty quality generally decrease as the

noise level increases. Overall, the student tends to outperform the teacher slightly. Building on the robustness study, we obtain an augmented student that outperforms the initial student in predictive performance, uncertainty quality, and calibration.

Regarding our final question of research, we hypothesized that the student is better able to detect OOD inputs. We find that the models vary in predictive uncertainty, and contrary to our expectations, the original student model does not consistently outperform the teacher in OOD uncertainty. However, the augmented student manages to outperform both the teacher and the original student in its ability to detect OOD inputs.

Therefore, in the context of our study, we successfully answered our research questions and obtained a detailed understanding of the uncertainty-aware distribution distillation framework when applied to a transformer-based model in the NLP domain.

Although our mix of approaches and the inherent novelty of our work makes direct comparisons difficult, we can compare trends and general observations to those of related studies. Our findings concerning the patterns of predictive performance, runtime, and uncertainty quality of teacher and student are consistent with those of Shen et al. (2021) who introduced the uncertainty-aware distribution distillation framework. In line with their work, the original student is more robust to noise than the teacher. Furthermore, we are able to successfully apply their proposal to augment the distillation process and obtain an improved student model. Finally, the results of our OOD analysis align well with the findings of Ovadia et al. (2019) and Shen et al. (2021), both regarding the original student and the augmented student.

This work combines different existing approaches from the uncertainty quantification and text classification spheres, applying knowledge from the CV domain to a common NLP task. Our results highlight the potential of using the uncertainty-aware distribution distillation framework for BERT for text classification tasks. As the framework can be easily adapted and applied to other transformer-based architectures, the findings of this work have broader implications. Our work is highly relevant to practical applications; their significant computational overhead often prevents the application of uncertainty quantification methods in practice, whereas uncertainty-aware distribution distillation enables obtaining high-quality real-time uncertainty estimates.

Although our study provides valuable insights into the application of a novel uncertainty quantification framework to the NLP domain, it is crucial to recognize its limitations. These include methodological shortcomings, such as the noise types used in the robustness study affecting the semantic content of the input sequences, unlike typical CV augmentations, which merely affect image properties, resulting in unnatural covariate shifts of the perturbed data. In addition, some of the observed result patterns, especially if they concern marginal differences in performance, are heavily influenced by the choices for model hyperparameters. Throughout our study, we aim to mitigate this potential issue by averaging the results of all experiments over multiple trials to maximize the robustness of our results. However, they are affected by various factors, and therefore, we should interpret the results of this study with caution. Moreover, it is worth noting that we observe a significantly smaller absolute inference overhead for MC dropout inference compared to CV applications such as Shen et al. (2021). This is due to the small input size of our hate speech detection dataset. The advantage of the framework is likely to be more emphasized for other types of text data that feature larger individual inputs. More generally, assessing reliable uncertainty estimates in NLP is challenging due to the discrete nature of language (Van Landeghem et al., 2022). A limitation for practical applications of our work might be posed by the scaling factor of the transfer dataset, which is heavily influenced by the specified sampling parameters. This may be problematic for applications involving very large datasets, although it can be partially solved by performing the transfer sampling and training of the student model in smaller batches. Recognizing these constraints provides a foundation for further research in this field.

Future research could study the application of uncertainty-aware distribution distillation to models tackling other NLP tasks that could take even better advantage of its real-time uncertainty quantification capabilities and multi-class problem settings to validate the approach further. Finally, further studies could focus on applying the framework to significantly larger transformer-based language models to enable high-quality uncertainty quantification while achieving state-of-the-art results.

Our study successfully applied the uncertainty-aware distribution distillation framework to the BERT model for hate speech detection in the NLP domain. While the results largely confirm our hypotheses, showing that the student model can match or even exceed the teacher in predictive performance and uncertainty quality, we observe that the results vary under different noise and OOD scenarios. We note that caution is needed in interpreting these findings due to the impact of methodological factors such as noise types and model hyperparameters. Despite these limitations, our research highlights the framework's potential for practical applications, especially in providing real-time uncertainty estimates, which could be the focus of further investigations.

# 9 Conclusion

This study successfully applied the uncertainty-aware distribution distillation framework to the BERT architecture for a hate speech detection task. We obtained a student model that outperforms the teacher in the F1 score and ECE. This student additionally proved more robust to covariate shifts in the evaluation data. We derived an augmented student that outperformed the original student across most metrics, particularly in its ability to detect OOD inputs due to its increased uncertainty quality, in which it performed significantly better than the teacher.

This work demonstrates the effective adaptation and application of the uncertainty-aware distribution distillation framework, previously applied within the CV domain, to the NLP domain. It allows us to harness the power of transformer-based models such as BERT for text classification tasks, providing real-time uncertainty estimates while retaining predictive performance and uncertainty quality. This is particularly valuable for researchers and practitioners who require fast and reliable uncertainty estimates in operational settings.

Guided by our research questions, we thoroughly investigated the application of this framework to a BERT classification model and confirmed our findings via a robustness study and OOD analysis. Our findings confirm that we successfully addressed our research objectives. Further work is required to explore the robustness of these results and the broader applicability of this framework across different NLP tasks.

## References


Bahuleyan, H., Mou, L., Vechtomova, O., and Poupart, P. (2018). Variational Attention for Sequence-to-Sequence Models. In Bender, E. M., Derczynski, L., and Isabelle, P., editors, *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1672–1682, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Barral Martínez, M. (2023). Platform regulation, content moderation, and AI-based filtering tools: Some reflections from the European Union. *Journal of Intellectual Property, Information Technology and Electronic Commerce Law*, 14(1):211.

Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Rangel Pardo, F. M., Rosso, P., and Sanguinetti, M. (2019). SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter. In May, J., Shutova, E., Herbelot, A., Zhu, X., Apidianaki, M., and Mohammad, S. M., editors, *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Belinkov, Y. and Bisk, Y. (2018). Synthetic and Natural Noise Both Break Neural Machine Translation. arXiv preprint arXiv:1711.02173.

Binns, R. (2018). Fairness in Machine Learning: Lessons from Political Philosophy. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, pages 149–159. PMLR.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight Uncertainty in Neural Network. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1613–1622. PMLR.

Bucila, C., Caruana, R., and Niculescu-Mizil, A. (2006). Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, page 535–541, New York, NY, USA. Association for Computing Machinery.

Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. (2020). ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. arXiv preprint arXiv:2003.10555.

Davidson, T., Warmsley, D., Macy, M., and Weber, I. (2017). Automated Hate Speech Detection and the Problem of Offensive Language. *Proceedings of the International AAAI Conference on Web and Social Media*, 11(1):512–515.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

Farquhar, S. (2022). *Understanding Approximation for Bayesian Inference in Neural Networks*. PhD thesis, University of Oxford.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874.

Gal, Y. (2016). *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1050–1059. PMLR.

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):44:1–44:37.

Georgakopoulos, S. V., Tasoulis, S. K., Vrahatis, A. G., and Plagianakos, V. P. (2018). Convolutional Neural Networks for Toxic Comment Classification. In *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*, SETN '18, pages 1–6, New York, NY, USA. Association for Computing Machinery.

Gillespie, T. (2020). Content moderation, AI, and the question of scale. *Big Data & Society*, 7(2).

Graves, A. (2011). Practical Variational Inference for Neural Networks. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1321–1330. PMLR.

Guo, H., Pasunuru, R., and Bansal, M. (2021). An Overview of Uncertainty Calibration for Text Classification and the Role of Distillation. In Rogers, A., Calixto, I., Vulić, I., Saphra, N., Kassner, N., Camburu, O.-M., Bansal, T., and Shwartz, V., editors, *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 289–306, Online. Association for Computational Linguistics.

He, J., Zhang, X., Lei, S., Chen, Z., Chen, F., Alhamadani, A., Xiao, B., and Lu, C. (2020). Towards More Accurate Uncertainty Estimation In Text Classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8362–8372, Online. Association for Computational Linguistics.

Hinton, G. and Van Camp, D. (1993). Keeping the neural networks simple by minimizing the description length of the weights. *Proceedings of the sixth annual conference on Computational learning theory*, pages 5–13.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the Knowledge in a Neural Network. arXiv preprint arXiv:1503.02531.

Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. (2011). Bayesian Active Learning for Classification and Preference Learning. arXiv preprint arXiv:1112.5745.

Hu, S., Pezzotti, N., and Welling, M. (2021). Learning to Predict Error for MRI Reconstruction. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part III*, pages 604–613, Berlin, Heidelberg. Springer-Verlag.

Hu, Y. and Khan, L. (2021). Uncertainty-Aware Reliable Text Classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, pages 628–636, New York, NY, USA. Association for Computing Machinery.

Hüllermeier, E. and Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506.

Izmailov, P., Nicholson, P., Lotfi, S., and Wilson, A. G. (2021a). Dangers of Bayesian Model Averaging under Covariate Shift: 35th Conference on Neural Information Processing Systems, NeurIPS 2021. *Advances in Neural Information Processing Systems 34 - 35th Conference on Neural Information Processing Systems, NeurIPS 2021*, pages 3309–3322.

Izmailov, P., Vikram, S., Hoffman, M. D., and Wilson, A. G. G. (2021b). What Are Bayesian Neural Network Posteriors Really Like? In *Proceedings of the 38th International Conference on Machine Learning*, pages 4629–4640. PMLR.

Jin, D., Jin, Z., Zhou, J. T., and Szolovits, P. (2020). Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8018–8025.

Kendall, A. and Gal, Y. (2017). What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., Lee, T., David, E., Stavness, I., Guo, W., Earnshaw, B., Haque, I., Beery, S. M., Leskovec, J., Kundaje, A., Pierson, E., Levine, S., Finn, C., and Liang, P. (2021). WILDS: A Benchmark of in-the-Wild Distribution Shifts. In *Proceedings of the 38th International Conference on Machine Learning*, pages 5637–5664. PMLR.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6405–6416, Red Hook, NY, USA. Curran Associates Inc.

Liang, B., Li, H., Su, M., Bian, P., Li, X., and Shi, W. (2018). Deep Text Classification Can be Fooled. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 4208–4215. International Joint Conferences on Artificial Intelligence Organization.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692.

MacKay, D. J. C. (1992). A Practical Bayesian Framework for Backpropagation Networks. *Neural Computation*, 4(3):448–472.

Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., and Wilson, A. G. (2019). A simple baseline for bayesian uncertainty in deep learning. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Malinin, A., Mlodozeniec, B., and Gales, M. (2019). Ensemble Distribution Distillation. arXiv preprint arXiv:1905.00076.

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., and Gao, J. (2021). Deep Learning–based Text Classification: A Comprehensive Review. *ACM Computing Surveys*, 54(3):62:1–62:40.

Miok, K., Škrlj, B., Zaharie, D., and Robnik-Šikonja, M. (2022). To BAN or Not to BAN: Bayesian Attention Networks for Reliable Hate Speech Detection. *Cognitive Computation*, 14(1):353–371.

Mozafari, M., Farahbakhsh, R., and Crespi, N. (2020). A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media. In Cherifi, H., Gaito, S., Mendes, J. F., Moro, E., and Rocha, L. M., editors, *Complex Networks and Their Applications VIII*, Studies in Computational Intelligence, pages 928–940, Cham. Springer International Publishing.

Mukherjee, S. and Awadallah, A. H. (2020). Distilling BERT into Simple Neural Networks with Unlabeled Transfer Data. arXiv preprint arXiv:1910.01769.

Murphy, A. H. (1973). A New Vector Partition of the Probability Score. *Journal of Applied Meteorology and Climatology*, 12(4):595–600.

Neal, R. M. (1993). Probabilistic Inference Using Markov Chain Monte Carlo Methods. *Technical Report CRG-TR*, 93(1).

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. Number 118 in Lecture Notes in Statistics. Springer.

Niculescu-Mizil, A. and Caruana, R. (2005). Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, pages 625–632, New York, NY, USA. Association for Computing Machinery.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. (2019). Can you trust your model' s uncertainty? Evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Postels, J., Ferroni, F., Coskun, H., Navab, N., and Tombari, F. (2019). Sampling-Free Epistemic Uncertainty Estimation Using Approximated Variance Propagation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2931–2940.

Pruksachatkun, Y., Krishna, S., Dhamala, J., Gupta, R., and Chang, K.-W. (2021). Does Robustness Improve Fairness? Approaching Fairness with Word Substitution Robustness Methods for Text Classification. arXiv preprint arXiv:2106.10826.

Quinonero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. (2022). *Dataset Shift in Machine Learning*. MIT Press.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving Language Understanding by Generative Pre-Training. OpenAI.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

Scheirer, W. J., de Rezende Rocha, A., Sapkota, A., and Boult, T. E. (2013). Toward Open Set Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1757–1772.

Shelmanov, A., Tsymbalov, E., Puzyrev, D., Fedyanin, K., Panchenko, A., and Panov, M. (2021). How Certain is Your Transformer? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1833–1840, Online. Association for Computational Linguistics.

Shen, Y., Zhang, Z., Sabuncu, M. R., and Sun, L. (2021). Real-Time Uncertainty Estimation in Computer Vision via Uncertainty-Aware Distribution Distillation. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 707–716, Waikoloa, HI, USA. IEEE.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Tang, R., Lu, Y., Liu, L., Mou, L., Vechtomova, O., and Lin, J. (2019). Distilling Task-Specific Knowledge from BERT into Simple Neural Networks. arXiv preprint arXiv:1903.12136.

Van Hee, C., Lefever, E., and Hoste, V. (2018). SemEval-2018 Task 3: Irony Detection in English Tweets. In Apidianaki, M., Mohammad, S. M., May, J., Shutova, E., Bethard, S., and Carpuat, M., editors, *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 39–50, New Orleans, Louisiana. Association for Computational Linguistics.

Van Landeghem, J., Blaschko, M., Anckaert, B., and Moens, M.-F. (2022). Benchmarking Scalable Predictive Uncertainty in Text Classification. *IEEE Access*, 10:43703–43737.

Varshney, K. R. and Alemzadeh, H. (2017). On the Safety of Machine Learning: Cyber-Physical Systems, Decision Sciences, and Data Products. arXiv preprint arXiv:1610.01256.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ukasz Kaiser, Ł., and Polosukhin, I. (2017). Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wang, Z. and Culotta, A. (2021). Robustness to Spurious Correlations in Text Classification via Automatically Generated Counterfactuals. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14024–14031.

Wei, J. and Zou, K. (2019). EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. arXiv preprint arXiv:1901.11196.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 681–688, Madison, WI, USA. Omnipress.

Wilson, A. G. (2021). Deep Ensembles as Approximate Bayesian Inference. https://cims.nyu.edu/~andrewgw/deepensembles/. Accessed 25.09.2023.

Wilson, A. G. and Izmailov, P. (2020). Bayesian Deep Learning and a Probabilistic Perspective of Generalization. In *Advances in Neural Information Processing Systems*, volume 33, pages 4697–4708. Curran Associates, Inc.

Yu, S., Chen, M., Zhang, E., Wu, J., Yu, H., Yang, Z., Ma, L., Gu, X., and Lu, W. (2020). Robustness study of noisy annotation in deep learning based medical image segmentation. *Physics in Medicine & Biology*, 65(17):175007.

## Declaration of Academic Honesty

I, Johann Benedikt Sonnenburg, hereby declare that I have not previously submitted the present work for other examinations. I wrote this work independently. All sources, including sources from the Internet, that I have reproduced in either an unaltered or modified form (particularly sources for texts, graphs, tables, and images), have been acknowledged by me as such. I understand that violations of these principles will result in proceedings regarding deception or attempted deception.

Hiermit erkläre ich, Johann Benedikt Sonnenburg, dass ich die vorliegende Arbeit nicht für andere Prüfungen eingereicht habe. Ich habe die Arbeit selbständig verfasst. Sämtliche Quellen einschließlich Internetquellen, die ich unverändert oder abgewandelt wiedergegeben habe, insbesondere Quellen für Texte, Grafiken, Tabellen und Bilder habe ich als solche kenntlich gemacht. Ich bin mir darüber bewusst, dass bei Verstößen gegen diese Grundsätze ein Verfahren wegen Täuschungsversuchs bzw. Täuschung eingeleitet wird.

Signature

---

Johann Benedikt Sonnenburg
Berlin, June 10, 2024