Bayesian Knowledge Distillation: A Bayesian Perspective of Distillation with Uncertainty Quantification

Luyang Fang [*1] **Yongkai Chen** [*1] **Wenxuan Zhong** [1] **Ping Ma** [1]

Abstract

Knowledge distillation (KD) has been widely used for model compression and deployment acceleration. Nonetheless, the statistical insight of the remarkable performance of KD remains elusive, and methods for evaluating the uncertainty of the distilled model/student model are lacking. To address these issues, we establish a close connection between KD and a Bayesian model. In particular, we develop an innovative method named *Bayesian Knowledge Distillation* (BKD) to provide a transparent interpretation of the working mechanism of KD, and a suite of Bayesian inference tools for the uncertainty quantification of the student model. In BKD, the regularization imposed by the teacher model in KD is formulated as a teacher-informed prior for the student model's parameters. Consequently, we establish the equivalence between minimizing the KD loss and estimating the posterior mode in BKD. Efficient Bayesian inference algorithms are developed based on the stochastic gradient Langevin Monte Carlo and examined with extensive experiments on uncertainty ranking and credible interval construction for predicted class probabilities.

1. Introduction

The exponential growth of parameters in deep learning models, driven by extensive resource allocation for training, has led to remarkable performance (Kondratyuk et al., 2023; Dosovitskiy et al., 2020; Fang et al., 2023). However, this growth also poses challenges in practical deployment due to the immense model sizes. Knowledge distillation (KD) emerges as an efficient solution for model compression designed to reduce the model size while maintaining performance, playing a pivotal role in addressing the challenges in model deployment (Zhao et al., 2022; Tung & Mori, 2019; Hinton et al., 2015; Huang & Wang, 2017; Li et al., 2020; Wang et al., 2023; Latif et al., 2023). Knowledge distillation (KD) is a technique where a smaller and simpler model (the student model) learns from a larger and more complex pre-trained model (the teacher model), similar to a student learning from a teacher. To facilitate this learning process, a penalty term is introduced that measures the dissimilarity between the predictions made by the teacher model and the student model. This penalty term is then incorporated into the loss function used for training the student model. By minimizing the combined loss, the student model is guided to make predictions that align with the teacher model's outputs. We refer to Gou et al. (2021) for a comprehensive survey on KD methods.

Despite the empirical success of KD, there remains a lack of clear statistical insight into the distillation process and its effects on the improvement of the student model. One commonly accepted intuition, as proposed by Hinton et al. (2015), is that the teacher model's prediction probabilities provide soft labels for the training data to the student model. Although these soft labels are more ambiguous than the true labels, they facilitate easier learning for the student model, thereby enhancing its performance. Numerous studies have provided a more comprehensive analysis to investigate the impact of distillation, as discussed in Section 2. However, it's worth noting that most existing research primarily concentrates on assessing how distillation enhances the prediction performance of the student model. While these insights are valuable, a deeper exploration of the statistical perspective and theoretical aspects of knowledge distillation is needed to gain a more comprehensive understanding of its efficacy and capability.

In this work, we approach this problem from a different perspective. We develop a novel method called Bayesian Knowledge Distillation (BKD) to distill knowledge from the teacher model to the student model in a Bayesian framework. Consider the classification task as an example. In BKD, the cross-entropy between prediction probabilities and labels of the data in the KD's loss function is treated as a likelihood function of the student model's parameters. A

[*]Equal contribution [1]Department of Statistics, University of Georgia, Athens, USA. Correspondence to: Wenxuan Zhong <wenxuan@uga.edu>, Ping Ma <pingma@uga.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

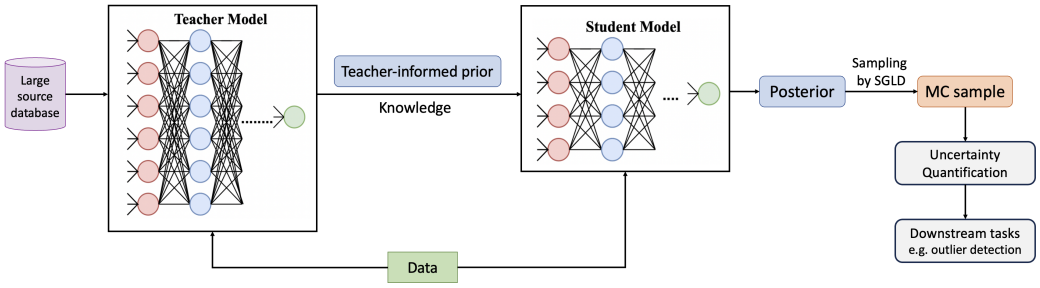

Figure 1. **Flowchart of Bayesian knowledge distillation.** We establish a teacher-informed prior for the student model's parameters based on the teacher model's predicted probabilities, and derive the posterior distribution. The stochastic Gradient Langevin Dynamics (SGLD) method is then applied to generate Monte Carlo samples from the posterior. Uncertainty quantification of the predictions and the subsequent downstream tasks, such as outlier detection, can be achieved accordingly.

teacher-informed prior, determined by the prediction probabilities provided by the teacher model, is specified for the parameters of the student model. This teacher-informed prior is then integrated with the likelihood to derive the posterior distribution of the student model's parameters. In such a formulation, one key finding is that minimizing the KD loss is equivalent to estimating the mode of the Bayesian posterior distribution, providing a transparent interpretation of the working mechanism of KD. We then naturally develop a suite of Bayesian inference tools for uncertainty quantification of the student model by sampling from the derived posterior distribution on the student model's parameters. We apply stochastic gradient Langevin dynamics (SGLD) (Welling & Teh, 2011), a subsampling-based Markov chain Monte Carlo (MCMC) algorithm, to generate the posterior MC samples. SGLD integrates stochastic optimization techniques with Langevin dynamics (Girolami & Calderhead, 2011; Roberts & Stramer, 2002) to enhance parameter updates, making it well-suited for high-dimensional, large-scale data (Ahn et al., 2012; Girolami & Calderhead, 2011; Teh et al., 2016). Specifically, deviance (Hastie, 1987) is utilized to quantify uncertainty, and the credible interval for it is established. Beyond providing conceptual insights, the new perspective offered by BKD facilitates novel applications of KD to challenges such as outlier detection and uncertainty ranking, leading to more robust decision-making.

We evaluate the proposed BKD on both synthetic and real benchmark datasets. We observe that the BKD method exhibits an increase in prediction uncertainty when faced with adversarial images generated from the original authentic dataset. This behavior suggests that these adversarial images fall in the tail regions of the training data distribution, highlighting the model's capacity to recognize inputs that fall outside its learned knowledge boundaries. The reliability of the provided uncertainty measures and the robust coverage rate of the credible intervals consistently underscore the effectiveness of our BKD method.

Our contributions are summarized as follows:

- We develop a novel BKD method that distills the teacher model into a compact student model by establishing a teacher-informed prior for the student model's parameters, along with a suite of Bayesian inference tools for uncertainty quantification. This approach provides a comprehensive framework for real-world applications such as outlier detection.
- We provide a transparent interpretation of the working mechanism of KD by establishing the equivalence between minimizing the KD loss and finding the posterior mode in BKD. This insight explains why KD can aid performance, offering a new perspective to improving KD methods.
- We showcase the capability of BKD for both improving the student model performance and enabling the uncertainty quantification of the prediction outcomes. The empirical performance of BKD is demonstrated on both synthetic and real datasets.

2. Related Research

The understanding of KD has been explored from several angles. Phuong & Lampert (2019) focus on the special scenarios where models are either linear or deep linear and prove a generalization bound that establishes extremely fast convergence of the risk of distillation-trained classifiers. Distillation has also been seen as a label smoothing regularization for the student model, and this idea has been widely explored in self-distillation (SD) (Yuan et al., 2020; Wang et al., 2021; Shen et al., 2022; Yun et al., 2020; Kim et al., 2021), a special case of KD where the teacher and student model share the same parameter space. Furthermore, Mobahi et al. (2020); Borup & Andersen (2021) interpret SD as obtaining the regularized parameter estimation, with

the teacher model defining the regularization term within the kernel ridge regression framework. In the setting of the Gaussian process model, Borup & Andersen (2023) proposes a distribution-centric SD approach. This approach shares a similar idea with BKD as it defines the prior of the student model with the teacher model. Nevertheless, given the divergence between SD and KD, these works focus on understanding how the student model can outperform the teacher model instead of analyzing the effectiveness of model compression.

Menon et al. (2021) provide a statistical perspective on knowledge distillation, explaining its effectiveness by presenting the knowledge of the teacher model as Bayesian prediction probabilities. They prove that KD can lower the prediction variance of the student model, and thus improve the performance. Although this study explains the teacher model as approximating Bayes probabilities and gives a concrete criterion for assessing a teacher model's performance, thereby leading to a more accurate student model, it falls short of quantifying the uncertainty associated with the student model. Phuong & Lampert (2019) focus on the distillation when the teacher and student models are simple linear and deep linear models. Under this special case, a generalization bound on the expected risk of the student model's prediction has been derived, and it is shown that the student model can almost perfectly mimic the teacher model's prediction with a fast convergence rate. Korattikara Balan et al. (2015) introduce *Bayesian Dark Knowledge* for distilling the posterior predictive distribution of the teacher model into a compact student model, thereby enabling the student model to quantify uncertainty. Subsequent extensions include Wang et al. (2018); Malinin et al. (2019); Vadera et al. (2020). However, these methods require access to the posterior predictive distribution of the teacher model, which in general is unavailable for many large pre-trained teacher models.

Another related problem is using a trained simpler model to stabilize the estimation of a complex model when sample sizes are too small. Huang et al. (2020) propose a *catalytic prior* and conducts Bayesian inference for the high-dimensional generalized linear model.

3. Preliminaries

We first introduce preliminaries about neural network models and knowledge distillation.

Neural Network Model. For the convenience of presentation, we introduce our method in the context of classification problems. In the classification tasks, we are given the training sample $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^m$ is the predictor and $y_i \in \{1, \ldots, K\}$ is the label of the i^{th} data point. Each data point is independently collected across $i \in \{1, \ldots, N\}$. We assume that the conditional probability of y_i is given by $\mathbb{P}(y_i = k|\mathbf{x} = \mathbf{x}_i)$ for each $k \in \{1, \ldots, K\}$.

The deep neural network (DNN) approximates the conditional probability function $\mathbb{P}(y = k|\mathbf{x})$ by applying a softmax transformation to the composition of a series of simple nonlinear functions. Without loss of generality, the approximated conditional probability function is denoted by a K-dimensional probability vector function $\boldsymbol{h}(\mathbf{x}, \boldsymbol{\theta}) = (h_1(\mathbf{x}, \boldsymbol{\theta}), \ldots, h_K(\mathbf{x}, \boldsymbol{\theta}))^T$, where $\boldsymbol{\theta} \in \mathbb{R}^d$ represents the weight and the bias parameters, $h_k(\mathbf{x}, \boldsymbol{\theta}) \in (0, 1)$ and $\sum_{k=1}^K h_k(\mathbf{x}, \boldsymbol{\theta}) \equiv 1$. For each $\mathbf{x}_i$, the label for the class with the highest probability in $\boldsymbol{h}(\mathbf{x}_i, \boldsymbol{\theta})$ is the predicted label.

As $\boldsymbol{h}(\cdot, \cdot)$ is known when the neural network structure is fixed, the parameter to be estimated is $\boldsymbol{\theta}$. In particular, we minimize the empirical risk

$$\mathcal{L}(\boldsymbol{h}(\cdot, \boldsymbol{\theta}); \mathcal{D}) = \frac{1}{N} \sum_{i=1}^N \text{CE}(\mathbf{y}_i, \boldsymbol{h}(\mathbf{x}_i, \boldsymbol{\theta})), \tag{1}$$

where $\mathbf{y}_i = (y_{i1}, \ldots, y_{iK})^T$ is the one-hot encoding of label y_i with $y_{ik} = \mathbf{1}_{y_i=k}$, and $\text{CE}(\mathbf{y}_i, \boldsymbol{h}(\mathbf{x}_i, \boldsymbol{\theta})) = -\sum_{k=1}^K y_{ik} \log\left(h_k(\mathbf{x}, \boldsymbol{\theta})\right)$ is the cross-entropy loss between the probability vector $\boldsymbol{h}(\mathbf{x}_i, \boldsymbol{\theta})$ and $\mathbf{y}_i$. Typically, the minimizer, denoted by $\boldsymbol{\theta}^*$, is found using the stochastic gradient descent (SGD) algorithm.

Knowledge Distillation. Knowledge distillation (KD), introduced by Hinton et al. (2015), is a procedure where a simpler model (student) learns from a larger model (teacher). In KD, we have a trained complex DNN model (teacher). For each data point $\mathbf{x}_i$ in the training sample $\mathcal{D}$, the teacher model predicts its class probability $\boldsymbol{p}_i = (p_{i1}, \ldots p_{iK})^T$, where p_{ij} is the predicted class probability that the i^{th} data point belongs to class j.

The student model M_s is trained using both the training sample $\mathcal{D}$ and the corresponding teacher model's predicted class probabilities $\boldsymbol{p} = \{\boldsymbol{p}_i\}_{i=1}^N$. As formulated before, M_s can be represented by $\boldsymbol{h}(\cdot, \boldsymbol{\theta})$. Given $\boldsymbol{p}$, KD measures the discrepancy between the student and teacher's model with

$$\begin{aligned} \tilde{\mathcal{L}}(\boldsymbol{h}(\cdot, \boldsymbol{\theta}); \mathcal{D}, \boldsymbol{p}) &= \frac{1}{N} \sum_{i=1}^N \text{CE}(\boldsymbol{p}_i, \boldsymbol{h}(\mathbf{x}_i, \boldsymbol{\theta})), \\ &= -\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K p_{ik} \log\left(h_k(\mathbf{x}_i; \boldsymbol{\theta})\right), \end{aligned} \tag{2}$$

which is the sample mean of the cross-entropy $\text{CE}(\boldsymbol{p}_i, \boldsymbol{h}(\mathbf{x}_i, \boldsymbol{\theta}))$ across the training data. To leverage the information from both the training data and the

teacher model's predictions, KD aims to solve

$$\begin{aligned}\boldsymbol{\theta}^*_{\mathrm{KD}} &= \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \left\{\mathcal{L}^{\mathrm{KD}}(\boldsymbol{h}(\cdot,\boldsymbol{\theta});\mathcal{D},\boldsymbol{p},\lambda)\right\} \\ &= \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \left\{\mathcal{L}(\boldsymbol{h}(\cdot,\boldsymbol{\theta});\mathcal{D}) + \lambda\tilde{\mathcal{L}}(\boldsymbol{h}(\cdot,\boldsymbol{\theta});\mathcal{D},\boldsymbol{p})\right\},\end{aligned} \tag{3}$$

where the minimized KD loss $\mathcal{L}^{\mathrm{KD}}(\boldsymbol{h}(\cdot,\boldsymbol{\theta});\mathcal{D},\boldsymbol{p},\lambda)$ is the linear combination of two loss terms in Equation (1) and Equation (2). λ is a constant factor weighting the contributions of these two terms of loss.

4. Bayesian Knowledge Distillation

4.1. Bayesian Model with a Teacher-Informed Prior

Let $\boldsymbol{q}_i = (q_{i1},\dots q_{iK})^T = \boldsymbol{h}(\mathbf{x}_i,\boldsymbol{\theta})$ denote the student model's predicted class probabilities for the i^{th} data point. In our proposed Bayesian model, the student model's parameter $\boldsymbol{\theta}$ is assumed to be random. Given the predicted class probabilities $\{\boldsymbol{p}_i\}_{i=1}^N$, where $\boldsymbol{p}_i = (p_{i1},\dots p_{iK})^T$, by the teacher model M_t, we aim to formulate a proper prior distribution $\pi_{\boldsymbol{\theta}}(\boldsymbol{\theta};\{\boldsymbol{p}_i\}_{i=1}^N)$, referred to as the teacher-informed prior (TIP), for $\boldsymbol{\theta}$.

To efficiently leverage the information provided by the teacher model's predicted class probabilities, we have the following assumption on the prior distribution $\pi_{\boldsymbol{\theta}}(\boldsymbol{\theta};\{\boldsymbol{p}_i\}_{i=1}^N)$,

$$\pi_{\boldsymbol{\theta}}(\boldsymbol{\theta};\{\boldsymbol{p}_i\}_{i=1}^N) \propto \Pi_{i=1}^N \pi_{\boldsymbol{q}}(\boldsymbol{q}=\boldsymbol{h}(\mathbf{x}_i,\boldsymbol{\theta});\boldsymbol{p}_i), \tag{4}$$

where $\propto$ denotes proportionality, and $\pi_{\boldsymbol{q}}(\boldsymbol{q};\boldsymbol{p}_i)$ is a probability density function defined for a K-dimensional probability vector $\boldsymbol{q} = (q_1,\dots,q_K)^T$ with the parameter $\boldsymbol{p}_i$. Equation (4) presents a readily analyzable structure for the prior of $\boldsymbol{\theta}$ by multiplying N well-defined probability functions $\pi_{\boldsymbol{q}}(\boldsymbol{q};\boldsymbol{p}_i)$. Following the same rationale as KD, the prior distribution $\pi_{\boldsymbol{\theta}}(\boldsymbol{\theta};\{\boldsymbol{p}_i\}_{i=1}^N)$ should assign a larger weight to the parameter $\boldsymbol{\theta}$ that results in the probability function $\pi_{\boldsymbol{q}}(\boldsymbol{q};\boldsymbol{p}_i)$ having a high probability around $\boldsymbol{p}_i$.

In this article, we consider $\pi_{\boldsymbol{q}}(\boldsymbol{q};\boldsymbol{p}_i)$ is the density function of a Dirichlet distribution $Dir(\mathbf{1}_K + \lambda\boldsymbol{p}_i)$, where λ is a tuning parameter gauging our confidence in the predicted probabilities of the teacher model M_t. Hence, we have

$$\pi_{\boldsymbol{q}}(\boldsymbol{q};\boldsymbol{p}_i) = \frac{1}{B(\mathbf{1}_K+\lambda\boldsymbol{p}_i)}\Pi_{k=1}^K (q_k)^{\lambda p_{ik}}, \tag{5}$$

where $B(\cdot)$ is the multivariate Beta function. Note that the mode of $\pi_{\boldsymbol{q}}(\boldsymbol{q};\boldsymbol{p}_i)$ is $\boldsymbol{p}_i$ for $\lambda > 0$. Furthermore, we show that $\pi_{\boldsymbol{\theta}}(\boldsymbol{\theta};\{\boldsymbol{p}_i\}_{i=1}^N)$ is a proper prior with the following proposition, whose proof is presented in Appendix B.

Proposition 4.1. *Consider the probability density function $\pi_{\boldsymbol{q}}(\boldsymbol{q};\boldsymbol{p}_i)$ as defined in Equation (5) with a constant $\lambda > 0$, assuming that the parameters of the student model lie in a compact space, then $\pi_{\boldsymbol{\theta}}(\boldsymbol{\theta};\{\boldsymbol{p}_i\}_{i=1}^N)$ is a proper prior.*

We now discuss the effect of λ. In the extreme case of $\lambda = 0$, $\pi_{\boldsymbol{q}}(\boldsymbol{q};\boldsymbol{p}_i)$ becomes the density of a symmetric Dirichlet distribution, $Dir(\mathbf{1}_K)$. This implies that there is no prior knowledge favoring one category over another when classifying each data point. For the case of $\lambda \to \infty$, we derive the following theorem to illustrate its influence,

Theorem 4.2. *Consider the probability density function $\pi_{\boldsymbol{q}}(\boldsymbol{q};\boldsymbol{p}_i)$ as defined in Equation (5), as $\lambda \to \infty$, we have*

$$\pi_{\boldsymbol{q}}(\boldsymbol{q};\boldsymbol{p}_i) \longrightarrow \delta(\boldsymbol{q}-\boldsymbol{p}_i), \tag{6}$$

where $\delta(\cdot)$ is the multivariate Dirac delta function.

The above theorem shows that, as λ approaches infinity, the student model's parameters only exhibit nonzero probability mass when the student model's predicted class probabilities are exactly equal to those of the teacher model. This result is consistent with the original KD where we have complete trust in the teacher model's class probabilities for infinitely large λ.

In summary, a large λ signifies our trust in the predicted distribution achieved by the teacher model. On the contrary, a small λ suggests that the knowledge provided by the teacher model are limited. In such cases, we would lean towards proposing a prior where the probabilities of each data point belonging to any specific class are approximately equal. Proof of Theorem 4.2 is available in Appendix B.

We further assume that $\mathbf{y}_i$ follows a multinomial distribution given the student model's parameter $\boldsymbol{\theta}$,

$$\mathbf{y}_i|\mathbf{x}_i \sim \text{Multinomial}(1;\boldsymbol{h}(\mathbf{x}_i,\boldsymbol{\theta})). \tag{7}$$

Consequently, we have the following theorem establishing the connection between our established Bayesian posterior and the original KD framework.

Theorem 4.3. *Given the prior distribution defined in Equation (4) and Equation (5), the posterior mode of $\boldsymbol{\theta}$ is the minimizer of KD, i.e., $\boldsymbol{\theta}^*_{KD}$.*

Proof. The negative log-transformed posterior distribution of $\boldsymbol{\theta}$,

$$\begin{aligned}-l(\boldsymbol{\theta};\mathcal{D},\boldsymbol{p},\lambda,\boldsymbol{h}(\cdot,\cdot)\} &= -\sum_{i=1}^N\sum_{k=1}^K y_{ik}\log\left(h_k(\mathbf{x}_i,\boldsymbol{\theta})\right) \\ &\quad -\lambda\sum_{i=1}^N\sum_{k=1}^K p_{ik}\log\left(h_k(\mathbf{x}_i,\boldsymbol{\theta})\right) + c, \\ &= \mathcal{L}^{\mathrm{KD}}(\boldsymbol{h}(\cdot,\boldsymbol{\theta});\mathcal{D},\boldsymbol{p},\lambda) + c,\end{aligned} \tag{8}$$

where c is a constant. It is easy to verify that the posterior mode of $\boldsymbol{\theta}$ is the minimizer of the KD loss in Equation (3). □

4.2. Bayesian inference for Prediction Uncertainty Quantification

Point prediction alone is often insufficient in practical applications, as it lacks an assessment of the prediction precision. In this subsection, we will show how our proposed Bayesian model is utilized in quantifying the prediction uncertainty.

Considering a new dataset $\mathcal{T}^n = \{(\tilde{\mathbf{x}}_i, \tilde{y}_i)\}_{i=1}^n$, with $\tilde{\mathbf{x}}_i \in \mathbb{R}^m$ and $\tilde{y}_i \in \{1, \ldots, K\}$, our objective is to perform predictions and conduct Bayesian inference on these predictions. Bayesian inference can be conducted naturally using the posterior distribution of $\boldsymbol{\theta}$, as derived in Equation (8). Nevertheless, for the high dimension of the parameter $\boldsymbol{\theta}$ and the complex structure of $\boldsymbol{h}(\cdot, \boldsymbol{\theta})$, it may not be feasible to obtain an analytical solution in these takes. Hence, we adopt the Langevin Monte Carlo method to tackle this problem.

Posterior Sampling. We apply the stochastic Gradient Langevin Dynamics (SGLD) (Welling & Teh, 2011), a variant of Langevin Monte Carlo, to efficiently sample from the complex, high-dimensional posterior distributions. SGLD achieves this by seamlessly integrating the general approach of stochastic gradient descent with Langevin dynamics.

Suppose $\boldsymbol{\theta}^{(j-1)}$ is sampled in iteration $j-1$. Then in the j-th step, given a mini-batch of m data points $\mathcal{D}^{(j)} = \{(\mathbf{x}_i^{(j)}, \mathbf{y}_i^{(j)})\}_{i=1}^m$ and the class probability $\{\boldsymbol{p}_i^{(j)}\}_{i=1}^m$ predicted by the teacher model, SGLD generates the sample from posterior using gradient updates plus Gaussian noise,

$$
\begin{aligned}
&\boldsymbol{\theta}^{(j)} \\
&= \boldsymbol{\theta}^{(j-1)} + \tau \nabla_{\boldsymbol{\theta}} l\left(\boldsymbol{\theta}^{(j-1)}; \mathcal{D}^{(j)}, \{\boldsymbol{p}_i^{(j)}\}_{i=1}^m, \lambda, \boldsymbol{h}\right) + \sqrt{2\tau}\xi^{(j)} \\
&= \boldsymbol{\theta}^{(j-1)} - \tau \nabla_{\boldsymbol{\theta}} \mathcal{L}^{\mathrm{KD}}\left(\boldsymbol{h}(\cdot, \boldsymbol{\theta}); \mathcal{D}, \boldsymbol{p}, \lambda\right) + \sqrt{2\tau}\xi^{(j)} \\
&= \boldsymbol{\theta}^{(j-1)} + \tau \sum_{i=1}^{m} \sum_{k=1}^{K} \frac{\mathbf{y}_i^{(j)} + \lambda \boldsymbol{p}_i^{(j)}}{h_k(\mathbf{x}_i^{(j)}, \boldsymbol{\theta})} \nabla_{\boldsymbol{\theta}} h_k(\mathbf{x}_i^{(j)}, \boldsymbol{\theta}) + \sqrt{2\tau}\xi^{(j)},
\end{aligned}
\tag{9}
$$

where τ is the step size, and $\xi^{(j)}$ is randomly sampled from $N(0, I)$. SGD optimizes the log-likelihood to guide the sampling process toward regions of higher probability, while Langevin dynamics introduces controlled noise into the parameter updates, ensuring convergence to the full posterior distribution rather than merely its mode. Specifically, in the limit of $j \to \infty$ and $\tau \to 0$, the probability distribution of $\boldsymbol{\theta}^{(j)}$, denoted as $\rho^{(j)}$, converges to a stationary distribution π, where π represents the distribution of $\boldsymbol{\theta}$. By using gradient information and introducing controlled noise, SGLD becomes more efficient in handling high-dimensional data (Girolami & Calderhead, 2011). Furthermore, SGLD's use of mini-batches for gradient computation alleviates the computational burden associated with optimizing over entire datasets, making it particularly well-suited for large-scale datasets. The BKD algorithm is summarized in Algorithm 1.

Algorithm 1 Bayesian Knowledge Distillation (BKD).

Input: $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, $\boldsymbol{h}(\cdot, \cdot)$, τ, λ, r.

1: Get the output $\boldsymbol{p}$ of the teacher model for each data point in $\mathcal{D}$.

2: Calculate the posterior distribution of $\boldsymbol{q} = \boldsymbol{h}(\mathbf{x}, \boldsymbol{\theta})$.

3: Generate Monte Carlo sample of $\boldsymbol{\theta}$:

- At iteration j^{th} with a subset of m data points $\mathcal{D}^{(j)} = \{(\mathbf{x}_i^{(j)}, \mathbf{y}_i^{(j)})\}_{i=1}^m$,
  - Generate $\xi^{(j)} \sim N(0, I)$,
  - Generate $\boldsymbol{\theta}^{(j)}$ using SGLD as in Equation (9).

Output: Monte Carlo sample $\{\boldsymbol{\theta}^{(j)}\}_{j=1}^r$ of $\boldsymbol{\theta}$.

Since the MC sample $\{\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots\}$ converges to the posterior distribution of $\boldsymbol{\theta}$, it allows for a precise estimation about the characteristics of $\boldsymbol{\theta}$. Consequently, it fosters a comprehensive analysis and understanding of our model covering various aspects. One aspect that we are particularly interested in is the model's prediction on the new dataset. Considering a new dataset $\mathcal{T}^n = \{(\tilde{\mathbf{x}}_i, \tilde{y}_i)\}_{i=1}^n$, with $\tilde{\mathbf{x}}_i \in \mathbb{R}^m$ and $\tilde{y}_i \in \{1, \ldots, K\}$, the posterior distribution of the model's predicted class probability $\boldsymbol{q}_i$ for the i^{th} data point can be approximated by MC sample $\{\widehat{\boldsymbol{q}}_i^{(j)} = \boldsymbol{h}(\tilde{\mathbf{x}}_i, \boldsymbol{\theta}^{(j)})\}_{j=1}^r$. Consequently, we can estimate the target characteristics of $\boldsymbol{q}_i$, such as posterior mode.

Measurement of Prediction Uncertainty. For each data point $(\tilde{\mathbf{x}}_i, \tilde{y}_i)$, we denote the model's prediction as $\tilde{\boldsymbol{q}}_i = \boldsymbol{h}(\tilde{\mathbf{x}}_1, \boldsymbol{\theta})$. We utilize deviance, which is a commonly used criterion, to measure the model performance and derive mean deviance as a metric to quantify uncertainty in predictions. For the multinomial distribution with observation $(\tilde{\mathbf{x}}_i, \tilde{y}_i)$, deviance has the following form

$$
\mathrm{dev}\left(\tilde{\mathbf{y}}_i, \tilde{\boldsymbol{q}}_i\right) = -2 \sum_{k=1}^{K} \tilde{y}_{ik} \log(\tilde{q}_{ik}), \tag{10}
$$

where $\tilde{\mathbf{y}}_i$ is the one-hot encoding for $\tilde{y}_i$. The expectation of $\mathrm{dev}\left(\tilde{\mathbf{y}}_i, \tilde{\boldsymbol{q}}_i\right)$ with respect to $\tilde{\mathbf{y}}_i$, written as $\Delta(\tilde{\boldsymbol{q}}_i)$, would be

$$
\Delta(\tilde{\boldsymbol{q}}_i) = E_{\mathbf{y} \sim Mul(1; \tilde{\boldsymbol{q}}_i)}\left(\mathrm{dev}(\mathbf{y}, \tilde{\boldsymbol{q}}_i)\right) = -2 \sum_{k=1}^{K} \tilde{q}_{ik} \log(\tilde{q}_{ik}). \tag{11}
$$

Here, $\Delta(\tilde{\boldsymbol{q}}_i)$ measures the generalized sum of squared residuals for the prediction $\tilde{\boldsymbol{q}}_i$. By taking the expectation over the distribution of $\tilde{\boldsymbol{q}}_i$, we obtain a measure of uncertainty for the prediction at the data point $(\tilde{\mathbf{x}}_i, \tilde{y}_i)$, which is expressed as $\overline{\Delta}_i$,

$$
\overline{\Delta}_i = E_{\boldsymbol{q}_i} \Delta(\boldsymbol{q}_i) = E_{\boldsymbol{q}_i} E_{\mathbf{y}}\left(\mathrm{dev}(\mathbf{y}, \boldsymbol{q}_i)\right). \tag{12}
$$

With the MC sample $\{\widehat{\boldsymbol{q}}_i^{(j)} = \boldsymbol{h}(\tilde{\mathbf{x}}_i, \boldsymbol{\theta}^{(j)})\}_{j=1}^r$ of $\tilde{\boldsymbol{q}}_i = \boldsymbol{h}(\tilde{\mathbf{x}}_i, \boldsymbol{\theta})$, we can estimate the prediction uncertainty $\overline{\Delta}_i$, by

simply taking the sample average, i.e.,

$$\overline{\Delta}_i = \frac{1}{r}\sum_{j=1}^{r}\Delta(\widehat{\boldsymbol{q}}_i^{(j)}) = \frac{1}{r}\sum_{j=1}^{r} E_{\mathbf{y}\sim Mul(1;\widehat{\boldsymbol{q}}_i^{(j)})}\left(\mathrm{dev}(\mathbf{y},\widehat{\boldsymbol{q}}_i^{(j)})\right)$$
$$= -\frac{2}{r}\sum_{j=1}^{r}\sum_{k=1}^{K}\widehat{q}_{ik}^{(j)}\log(\widehat{q}_{ik}^{(j)}). \quad (13)$$

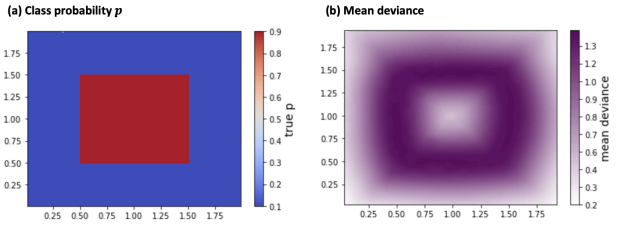

Figure 2. Results for the toy checkerboard example. The left panel presents the data points with the color determined by class probability p. The right panel presents the data points with the color determined by mean deviance.

We present a toy checkerboard example to illustrate the effectiveness of mean deviance in quantifying uncertainty. In this example, we define $p(x) = 0.9$ when $\frac{1}{2} < x_1 < \frac{3}{2}$ and $\frac{1}{2} < x_2 < \frac{3}{2}$, otherwise, $p(x)$ is set to 0.1. Here, we set $x_1 \sim Unif(0,2)$ and $x_2 \sim Unif(0,2)$. The left panel of Figure 2 displays the class probability p for each data point, and the right panel illustrates the predicted mean deviance for each data point. These results clearly demonstrate that the model exhibits higher prediction uncertainty when data points are surrounded by those from other classes. This phenomenon validates the ability of our model to effectively quantify uncertainty. Furthermore, this indicates that prediction uncertainty is influenced not only by class probability p but also by additional factors like neighborhood information. Consequently, relying solely on the posterior mode of p as a measure of uncertainty may be inadequate.

Credible Interval. Now we develop the construction of the credible interval for the deviance of the prediction. For each data point $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i)$, the $1-\alpha$ credible interval for its deviance, denoted by $CI_i = [0, \tau_i]$, satisfies

$$E_{\boldsymbol{q}_i}[E_{\mathbf{y}\sim Mul(1,\boldsymbol{q}_i)}(\mathbf{1}_{\{\mathrm{dev}(\mathbf{y},\boldsymbol{q}_i)\le\tau_i\}})] = 1-\alpha, \quad (14)$$

where α is the credible level. With the MC sample $\{\widehat{\boldsymbol{q}}_i^{(j)} = \boldsymbol{h}(\tilde{\mathbf{x}}_i, \boldsymbol{\theta}^{(j)})\}_{j=1}^{r}$, we can empirically estimate the optimal τ_i by minimizing $L(\tau_i)$ as defined as,

$$L(\tau_i) = \left|\frac{1}{r}\sum_{j=1}^{r} E_{\mathbf{y}\sim Mul(1;\widehat{\boldsymbol{q}}_i^{(j)})}(\mathbf{1}_{\{\mathrm{dev}(\mathbf{y},\widehat{\boldsymbol{q}}_i^{(j)})\le\tau_i\}}) - (1-\alpha)\right|$$
$$= \left|\frac{1}{r}\sum_{j=1}^{r}\Big(\widehat{\boldsymbol{q}}_i^{(j)}\mathbf{1}_{\{\mathrm{dev}(1,\widehat{\boldsymbol{q}}_i^{(j)})\le\tau_j\}} + (1-\widehat{\boldsymbol{q}}_i^{(j)})\mathbf{1}_{\{\mathrm{dev}(0,\widehat{\boldsymbol{q}}_i^{(j)})\le\tau_j\}}\Big) - (1-\alpha)\right|. \quad (15)$$

The function $L(\tau_i)$ denotes the difference between the left-hand side and the right-hand side of Equation (14).

Constructing a credible interval on deviance, rather than directly on the prediction $\boldsymbol{q}$, presents a notable advantage. It enables the calculation of the coverage rate of the constructed interval using the true label y. This is beneficial since, in most cases, the true labels of the testing dataset are readily available for evaluation, whereas the true class probabilities are typically elusive. Specifically, with the constructed credible interval $[0, \widehat{\tau}_i]$, the coverage rate $\widehat{R}_i$ for the i^{th} testing data point $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i)$ is estimated by

$$\widehat{R}_i = \frac{1}{r}\sum_{j=1}^{r}\mathbf{1}_{\left\{\mathrm{dev}\left(\tilde{\mathbf{y}}_i,\widehat{\boldsymbol{q}}_i^{(j)}\right)\le\widehat{\tau}_i\right\}}. \quad (16)$$

Consequently, the average coverage rate, denoted as $\widehat{R}$, of the testing dataset $\mathcal{T}^n$ is estimated by

$$\widehat{R} = \frac{1}{n}\sum_{i=1}^{n}\widehat{R}_i = \frac{1}{nr}\sum_{i=1}^{n}\sum_{j=1}^{r}\mathbf{1}_{\left\{\mathrm{dev}\left(\tilde{\mathbf{y}}_i,\widehat{\boldsymbol{q}}_i^{(j)}\right)\le\widehat{\tau}_i\right\}}. \quad (17)$$

5. Real Data Analysis

We test the proposed BKD method on four benchmark datasets, (1) MNIST, (2) Fashion MNIST, (3) CIFAR-10, and (4) CIFAR-100. Detailed information about the datasets can be found in Appendix D.1.

We compare BKD with four benchmark methods: (1) the teacher model; (2) the original KD method; (3) the integration of the original KD and Bayesian neural network (BNN) (Blundell et al., 2015); and (4) the integration of the original KD and Monte Carlo dropout (Dropout) (Gal & Ghahramani, 2016). Please refer to Appendix D.3 for details about the implementation of BNN and Dropout. We evaluate the performance of the BKD method in terms of classification accuracy and uncertainty quantification. We also evaluate BKD on some synthetic datasets, presented in Appendix C.

5.1. Classification Results

Table 1 details the teacher and student models that we use for each dataset, including the model structures and the number of parameters for each model in parentheses. The chosen student models are noticeably smaller in size when compared to their corresponding teacher models. For detailed information on the model structure, please refer to Appendix D.2.

Table 2 presents the classification accuracy of all methods on four benchmark datasets. Despite their smaller model sizes, student models trained with KD methods perform comparable to the teacher models, with the student models even having a higher accuracy than the teacher model on the Fashion MNIST dataset. These phenomenons highlight

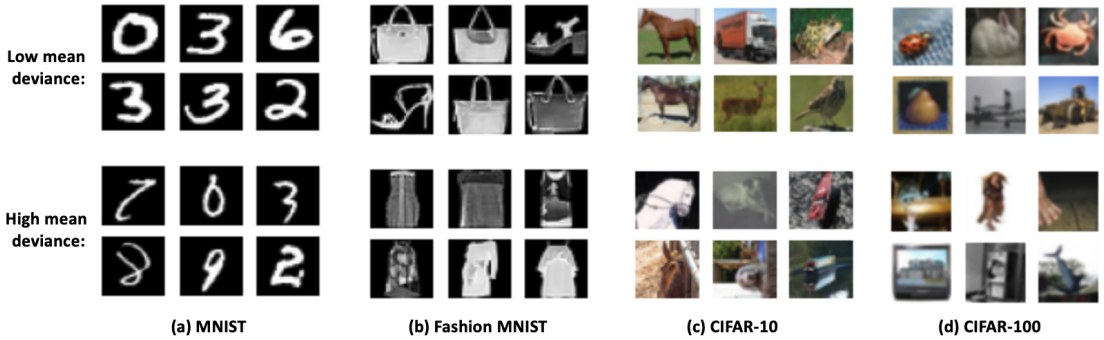

Figure 3. Top panel showcases 6 images with the lowest mean deviance, while the bottom panel showcases 6 images with the highest mean deviance, for (a) MNIST, (b) Fashion MNIST, (c) CIFAR-10, and (d) CIFAR-100 datasets separately.

Table 1. Model Structure. The number of parameters for each respective model is indicated in parentheses.

	Teacher Model	Student Model
MNIST	MLP-L (2.4M)	MLP-S (0.2M)
Fashion MNIST	ResNet-50 (25.6M)	CNN (0.08M)
CIFAR-10	ViT-B-16 (86M)	MUXNet-m (3.4M)
CIFAR-100	ViT-B-16 (86M)	MUXNet-m (3.4M)

the effectiveness of both the original KD method and BKD in distilling the knowledge into a compact student model without compromising performance. Across all datasets, the performance of all methods is comparable, with BNN demonstrating a slightly lower accuracy while BKD exhibits a slight improvement.

Table 2. Accuracy Results. Accuracy of five methods on four benchmark datasets separately.

	Accuracy				
	Teacher	Orig KD	BNN	Dropout	BKD
MNIST	0.990	0.986	0.979	0.984	0.984
F-MNIST	0.901	0.902	0.890	0.902	0.905
CIFAR-10	0.989	0.963	0.906	0.962	0.964
CIFAR-100	0.929	0.841	0.787	0.840	0.842

We further consider all KD methods' performance when taking the perturbed images as input. An effective model should exhibit adaptability by achieving higher accuracy and less uncertainty (indicated by reduced mean deviance in our case) for clean images while showing lower accuracy and greater uncertainty for noisy images. We generate perturbed images $\mathbf{x}_{\text{perturb}}$ from the original MNIST images $\mathbf{x}$ by setting

$$\mathbf{x}_{\text{perturb}} = 2\left(\frac{\mathbf{x}^* - \mathbf{x}^*_{\min}}{\mathbf{x}^*_{\max} - \mathbf{x}^*_{\min}}\right) - 1, \text{ with } \mathbf{x}^* = \mathbf{x} + \gamma\boldsymbol{\epsilon}, \quad (18)$$

where $\mathbf{x} \in \mathbb{R}^m$ is the original image, $\boldsymbol{\epsilon} \sim N(0, I_m)$ is the noise term, and γ is the perturbation level. As the perturbation level increases, we expect to observe a decrease in accuracy and an increase in mean deviance.

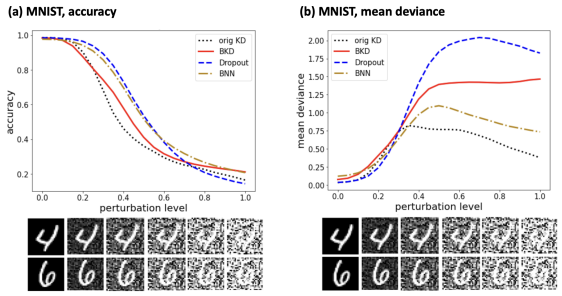

Figure 4. Classification accuracy and mean deviance results as a function of perturbation level for the MNIST dataset. Examples of perturbed images are shown below the axis. The red lines show the results of the proposed BKD method, while the blue line represents the original KD method.

The left panel of Figure 4 shows the classification accuracy on perturbed images, while the right panel shows the estimated mean deviance. Below the axis, examples of perturbed images with varying perturbation levels are presented. As expected, we observe a decrease in accuracy with increasing perturbation levels for all the methods. However, the trend in mean deviance as the perturbation level increases reveals a notable pattern for BKD compared to other methods. The original KD method, BNN, and Dropout tend to make overconfident predictions, while BKD provides more reasonable predictions. Specifically, at relatively high perturbation levels ($\gamma > 0.3$ for original KD, $\gamma > 0.5$ for BNN, and $\gamma > 0.7$ for Dropout), the accuracy and predictive deviance of those methods have both decreased, indicating high confidence in incorrect predictions. In contrast, BKD understands its inaccuracies at high perturbation levels by presenting high and increasing uncertainty estimates. These findings highlight that BKD offers a more accurate assessment of uncertainty compared to other methods.

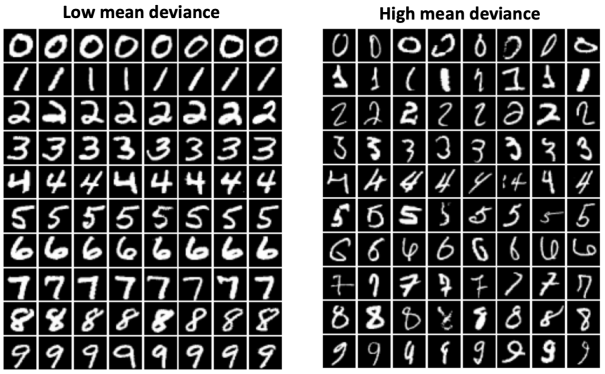

Figure 5. For each class in the MNIST dataset, the eight images with the lowest mean deviance (left), and the eight images with the highest mean deviance (right) are displayed.

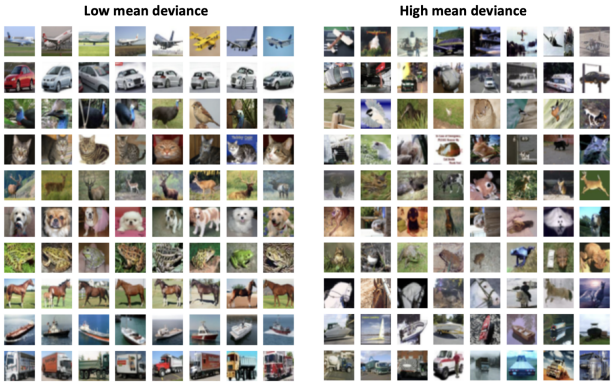

Figure 6. For each class in the CIFAR-10 dataset, the eight images with the lowest mean deviance are displayed on the left, while the eight images with the highest mean deviance are shown on the right.

5.2. Uncertainty Evaluation

Uncertainty. We use the mean deviance detailed in Equation (13) as the metric to quantify the model uncertainty for each image.

In Figure 3, images with the lowest and highest mean deviance are shown on the top and bottom panels, respectively, with each column dedicated to a specific dataset. The images with higher mean deviance in the bottom panel indicate that the model provides higher uncertainty regarding these predictions. These images are noticeably more difficult to classify. The clear association between high uncertainty and inherent difficulty in image recognition validates the performance of BKD in uncertainty quantification. We also present a separate visualization for each class within the MNIST dataset in Figure 5, and results for the CIFAR-10 dataset in Figure 6. We observe that images exhibiting greater mean deviance (right panel) tend to be more cluttered, with the subject being less prominent. This reasonably leads to large uncertainty in the model's predictions. Similar visualization results for other datasets are available in Appendix D.5.

To further evaluate the performance of BKD on uncertainty quantification, we explore the distribution of mean deviance over all images as well as the distribution of images in each class. Figure 7 shows the log-transformed mean deviance distribution for the Fashion MNIST dataset, with the first box representing results across all classes and subsequent boxes detailing results by class. Notably, classes 'trouser', 'bag', and 'sandal' exhibit lower mean deviance, reflecting their distinctive and recognizable features. Conversely, classes 'T-shirt', 'pullover', 'coat', and 'shirt' show higher mean deviance, underscoring the challenges in differentiating these items. The observed pattern is expected because items with distinctive designs, such as the specific length of trousers or the particular shape of bags and sandals, tend to be identified more easily. In contrast, upper-body garments create a tougher task for recognition due to their generally similar outlines, and the fine details are not always clear, especially in low-resolution images. Similar analyses for other datasets can be found in Appendix D.5.

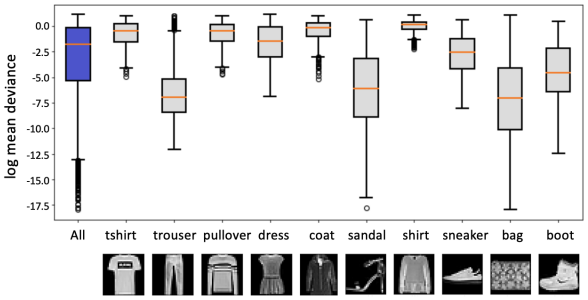

Figure 7. The boxplot illustrates the log-transformed mean deviance for the Fashion MNIST dataset. The leftmost box shows the result for all images, whereas the following boxes detail the results for each class.

Coverage rate. In Figure 8, we report the coverage rates at the commonly used 95% credible level for the testing datasets with different sizes (500, 1000, 2000, 4000, 6000, 8000, 10000) on the CIFAR-100 dataset. The results are based on 10 repetitions and consistently demonstrate that our BKD method achieves coverage rates closely matching the specified 95% credible level. As the sample size of the data increases, variation across repetitions decreases, and the mean coverage rate stabilizes near 0.95. Results on the other three datasets are reported in Figure 16 in Appendix D.5.

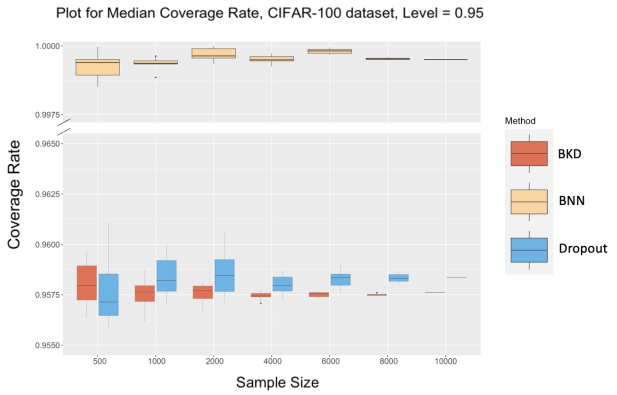

Figure 8. Evaluation of coverage rate on commonly used credible level 0.95 for various sample sizes in the CIFAR-100 dataset. We compare the results of BKD, BNN, and Dropout.

6. Conclusion

In this work, we develop a Bayesian Knowledge Distillation (BKD) method aimed at compressing the teacher model and providing a suite of Bayesian inference tools for the student model's uncertainty quantification. BKD establishes a teacher-informed prior for the student model's parameters, from which the posterior is subsequently derived. We show that minimizing KD loss is tantamount to estimating the posterior mode in BKD, thereby offering a clear interpretation of KD's operational mechanism and suggesting novel ways to improve KD methods. Furthermore, we establish the theoretical properties of BKD. To handle high-dimensional, large-scale data, we apply stochastic gradient Langevin dynamics for generating posterior samples.

We evaluate the performance of BKD using both synthetic and real datasets, focusing on classification accuracy and uncertainty quantification ability. The results demonstrate that BKD matches the teacher model's performance with the advantage of a significantly smaller size. The key strength of BKD primarily lies in precisely quantifying prediction uncertainty, indicated by three key findings. First, we show that mean deviance is validated as a reliable uncertainty metric by exploring its association with both the true class probability p and the intrinsic structure of data points (Figure 2). Second, our analysis of real datasets reveals a tendency of the original KD method to yield overly confident predictions when applied to perturbed images. In contrast, BKD consistently provides predictions with reasonable uncertainty. Third, the visualization and distribution analysis of estimated uncertainty across classes confirms the reliability of BKD's uncertainty measures. Additionally, the exploration of coverage rates demonstrates the robustness of BKD.

There are some potential extension directions of BKD of great interest. First, we may employ priors other than the Dirichlet distribution for the student model's parameters. We use Dirichlet distribution to achieve one of the primary goals of this manuscript, i.e., interpreting the original KD. However, alternative priors, such as the continuous categorical (CC) distribution, could be considered based on the practical context. In Appendix D.4, we provide the experiment results evaluating the performance of BKD on the MNIST dataset using CC prior.

Second, we may impose a more specific and restricted student model, with a focus on the direct inference of its parameters. In this way, BKD may provide an analytical form of the posterior distribution of these parameters. For example, if the linear discriminant analysis (LDA) classifier is used in the student model. Utilizing BKD with our currently defined prior in Equation (5), the posterior distribution of class means and covariance matrix leads to the normal-inverse-Wishart distribution. A simulation study setting QDA as the teacher model and LDA as the student model is reported in Appendix C.5.

Third, BKD could potentially be extended to regression and data generation tasks. To achieve this objective, the key issue is to impose a suitable prior for the student model. For regression, the teacher-informed prior distribution can be constructed as the Gaussian density function with the mean as the prediction results of the teacher model. For sore-based generative models, we can establish the prior for the score function as the Gaussian density function with the means as the score functions estimated by the teacher model using the training data.

This work contributes to the growing demand for reliable, privacy-aware, and resource-efficient machine learning models in real-world applications.

Acknowledgements

This work was partially supported by the U.S. National Science Foundation [DMS-1925066, DMS-1903226, DMS-2124493, DMS-2311297, DMS-2319279, DMS-2318809] and the National Institutes of Health [NIH R01GM152814].

Impact Statement

This study stays at the intersection of statistics and computer science. The developed BKD method is expected to bridge the void in deploying pre-trained super large deep learning models safely, of which the potential impact on areas where data privacy is of significant importance, such as the medical AI area, is enormous. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Ahn, S., Korattikara, A., and Welling, M. Bayesian posterior sampling via stochastic gradient fisher scoring. *arXiv preprint arXiv:1206.6380*, 2012.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural network. In *International conference on machine learning*, pp. 1613–1622. PMLR, 2015.

Borup, K. and Andersen, L. N. Even your teacher needs guidance: Ground-truth targets dampen regularization imposed by self-distillation. *Advances in Neural Information Processing Systems*, 34:5316–5327, 2021.

Borup, K. and Andersen, L. N. Self-distillation for Gaussian process regression and classification. *arXiv preprint arXiv:2304.02641*, 2023.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Fang, L., Lee, G.-G., and Zhai, X. Using GPT-4 to augment unbalanced data for automatic scoring. *arXiv preprint arXiv:2310.18365*, 2023.

Gal, Y. and Ghahramani, Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Girolami, M. and Calderhead, B. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 73(2):123–214, 2011.

Gordon-Rodriguez, E., Loaiza-Ganem, G., and Cunningham, J. The continuous categorical: a novel simplex-valued exponential family. In *International Conference on Machine Learning*, pp. 3637–3647. PMLR, 2020.

Gou, J., Yu, B., Maybank, S. J., and Tao, D. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

Hastie, T. A closer look at the deviance. *The American Statistician*, 41(1):16–20, 1987.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hinton, G., Vinyals, O., Dean, J., et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Huang, D., Stein, N., Rubin, D. B., and Kou, S. Catalytic prior distributions with application to generalized linear models. *Proceedings of the National Academy of Sciences*, 117(22):12004–12010, 2020.

Huang, Z. and Wang, N. Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219*, 2017.

Kim, K., Ji, B., Yoon, D., and Hwang, S. Self-knowledge distillation with progressive refinement of targets. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6567–6576, 2021.

Kondratyuk, D., Yu, L., Gu, X., Lezama, J., Huang, J., Hornung, R., Adam, H., Akbari, H., Alon, Y., Birodkar, V., et al. Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*, 2023.

Korattikara Balan, A., Rathod, V., Murphy, K. P., and Welling, M. Bayesian dark knowledge. *Advances in neural information processing systems*, 28, 2015.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Latif, E., Fang, L., Ma, P., and Zhai, X. Knowledge distillation of LLM for education. *arXiv preprint arXiv:2312.15842*, 2023.

LeCun, Y. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Li, T., Li, J., Liu, Z., and Zhang, C. Few sample knowledge distillation for efficient network compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14639–14647, 2020.

Lu, Z., Deb, K., and Boddeti, V. N. Muxconv: Information multiplexing in convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12044–12053, 2020.

Malinin, A., Mlodozeniec, B., and Gales, M. Ensemble distribution distillation. *arXiv preprint arXiv:1905.00076*, 2019.

Menon, A. K., Rawat, A. S., Reddi, S., Kim, S., and Kumar, S. A statistical perspective on distillation. In *International Conference on Machine Learning*, pp. 7632–7642. PMLR, 2021.

Mobahi, H., Farajtabar, M., and Bartlett, P. Self-distillation amplifies regularization in Hilbert space. *Advances in Neural Information Processing Systems*, 33:3351–3361, 2020.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Phuong, M. and Lampert, C. Towards understanding knowledge distillation. In *International conference on machine learning*, pp. 5142–5151. PMLR, 2019.

Roberts, G. O. and Stramer, O. Langevin diffusions and Metropolis-Hastings algorithms. *Methodology and computing in applied probability*, 4:337–357, 2002.

Shen, Y., Xu, L., Yang, Y., Li, Y., and Guo, Y. Self-distillation from the last mini-batch for consistency regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11943–11952, 2022.

Shridhar, K., Laumann, F., and Liwicki, M. A comprehensive guide to Bayesian convolutional neural network with variational inference. arxiv 2019. *arXiv preprint arXiv:1901.02731*, 2019.

Teh, Y., Thiéry, A., and Vollmer, S. Consistency and fluctuations for stochastic gradient langevin dynamics. *Journal of Machine Learning Research*, 17, 2016.

Tung, F. and Mori, G. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1365–1374, 2019.

Vadera, M., Jalaian, B., and Marlin, B. Generalized Bayesian posterior expectation distillation for deep neural networks. In *Conference on Uncertainty in Artificial Intelligence*, pp. 719–728. PMLR, 2020.

Van der Maaten, L. and Hinton, G. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11), 2008.

Wang, H., Li, Y., Xu, W., Li, R., Zhan, Y., and Zeng, Z. Dafkd: Domain-aware federated knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20412–20421, 2023.

Wang, K.-C., Vicol, P., Lucas, J., Gu, L., Grosse, R., and Zemel, R. Adversarial distillation of Bayesian neural network posteriors. In *International conference on machine learning*, pp. 5190–5199. PMLR, 2018.

Wang, Y., Li, H., Chau, L.-p., and Kot, A. C. Embracing the dark knowledge: Domain generalization using regularized knowledge distillation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 2595–2604, 2021.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688, 2011.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Yuan, L., Tay, F. E., Li, G., Wang, T., and Feng, J. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3903–3911, 2020.

Yun, S., Park, J., Lee, K., and Shin, J. Regularizing class-wise predictions via self-knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13876–13885, 2020.

Zhao, B., Cui, Q., Song, R., Qiu, Y., and Liang, J. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 11953–11962, 2022.

APPENDIX

Here is an outline of the Appendix.

- Appendix A summarizes the notations used in the main text.
- Appendix B gives the proof of theoretical results.
- Appendix C presents the simulation experiments.
- Appendix D presents supplementary details for real data analysis.

A. Notation

In Table 3, we summarize the notations in the main text.

Notation	Interpretation
$\mathcal{F}$	function classes
$\mathbb{P}$	population distribution
$\mathcal{D}, \mathcal{T}$	dataset
$\mathcal{P}$	distribution class
$\boldsymbol{h}$	function representing the student neural network
$\boldsymbol{\theta}$	parameters of the student neural network
$\boldsymbol{\theta}^*$	estimate of $\boldsymbol{\theta}$
$\widehat{\boldsymbol{\theta}}$	Monte Carlo sample of $\boldsymbol{\theta}$
$\mathbf{x}$	vector representing m-dimensional feature
y	class label
$\mathbf{y}$	one-hot encoding of class label y
$\boldsymbol{p}$	output of the tea neural network, representing the predicted class probability
$\boldsymbol{q}$	output of the student neural network, representing the predicted class probability
N	sample size of the training dataset $\mathcal{D}$
n	sample size of the testing dataset $\mathcal{T}$
l	likelihood function
∇	gradient operator
λ	weight parameter
dev	deviance

Table 3. Notation table

B. Proof of Theoretical Results

Proposition 4.1. *Consider the probability density function $\pi_{\boldsymbol{q}}(\boldsymbol{q}; \boldsymbol{p}_i)$ as defined in Equation (5) with a constant $\lambda > 0$. Under the condition that the parameters of the student model lie in a compact space, $\pi_{\boldsymbol{\theta}}(\boldsymbol{\theta}; \{\boldsymbol{p}_i\}_{i=1}^N)$ is a proper prior.*

Proof of Proposition 4.1. According to Equation (4) and (5), for some constant $C > 0$, we have

$$\begin{aligned} \pi_{\boldsymbol{\theta}}(\boldsymbol{\theta}; \{\boldsymbol{p}_i\}_{i=1}^N) &= C \, \Pi_{i=1}^N \pi_{\boldsymbol{q}}(\boldsymbol{q} = \boldsymbol{h}(\mathbf{x}_i, \boldsymbol{\theta}); \boldsymbol{p}_i) \\ &= C \, \Pi_{k=1}^K \Pi_{i=1}^N \left(\boldsymbol{h}(\mathbf{x}_i, \boldsymbol{\theta})\right)^{\lambda p_{ik}}. \end{aligned} \tag{B.1}$$

Taking integration, we get

$$\int \pi_{\boldsymbol{\theta}}(\boldsymbol{\theta}; \{\boldsymbol{p}_i\}_{i=1}^N) \mathrm{d}\boldsymbol{\theta} = C \int \Pi_{k=1}^K \Pi_{i=1}^N \left(\boldsymbol{h}(\mathbf{x}_i, \boldsymbol{\theta})\right)^{\lambda p_{ik}} \mathrm{d}\boldsymbol{\theta}. \tag{B.2}$$

Since $0 \leq \boldsymbol{h}(\mathbf{x}_i, \boldsymbol{\theta}) \leq 1$, we know $0 \leq \Pi_{k=1}^K \Pi_{i=1}^N \left(\boldsymbol{h}(\mathbf{x}_i, \boldsymbol{\theta})\right)^{\lambda p_{ik}} \leq 1$ for all i and k. Because we assume $\boldsymbol{\theta}$ is from a compact space and $\Pi_{k=1}^K \Pi_{i=1}^N \left(\boldsymbol{h}(\mathbf{x}_i, \boldsymbol{\theta})\right)^{\lambda p_{ik}}$ is bounded, we know the integration B.2 is bounded. Thus, we have shown $\pi_{\boldsymbol{\theta}}(\boldsymbol{\theta}; \{\boldsymbol{p}_i\}_{i=1}^N)$ is a proper prior. □

Theorem 4.2. *Consider the probability density function $\pi_{\boldsymbol{q}}(\boldsymbol{q};\boldsymbol{p}_i)$ as defined in Equation (5), as $\lambda \to \infty$, we have*

$$\pi_{\boldsymbol{q}}(\boldsymbol{q};\boldsymbol{p}_i) \longrightarrow \delta(\boldsymbol{q}-\boldsymbol{p}_i), \tag{B.3}$$

where $\delta(\cdot)$ is the multivariate Dirac delta function.

Proof of Theorem 4.2. For the sake of brevity, we consider the general formula $\pi_{\boldsymbol{q}}(\boldsymbol{q};\boldsymbol{p},\lambda)$, where $\boldsymbol{q}=(q_1,q_2,\cdots,p_K)^T$ and $\boldsymbol{p}=(p_1,p_2,\cdots,p_K)^T$. WLOG, we consider the situation where $p_k \neq 0$ for $k \in 1,\cdots,K$. We have

$$\pi_{\boldsymbol{q}}(\boldsymbol{q};\boldsymbol{p},\lambda) = \frac{1}{B(\mathbf{1}_K+\lambda\boldsymbol{p})}\Pi_{k=1}^K (q_k)^{\lambda p_k} = \frac{\Gamma(\lambda+K)}{\prod_{k=1}^K(\Gamma(\lambda p_k+1))}\Pi_{k=1}^K (q_k)^{\lambda p_k}. \tag{B.4}$$

We start with the situation when $\lambda p_k, k=1,\cdots,K$, are all integers. We have

$$\pi_{\boldsymbol{q}}(\boldsymbol{q};\boldsymbol{p},\lambda) = \frac{(\lambda+K-1)!}{\prod_{k=1}^K(\lambda p_k)!}\cdot\prod_{k=1}^K (q_k)^{\lambda p_k}. \tag{B.5}$$

According to the Stirling formula that $n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$, we have

$$\begin{aligned}\pi_{\boldsymbol{q}}(\boldsymbol{q};\boldsymbol{p},\lambda) &\sim \frac{\lambda^{K-1}\cdot\sqrt{\lambda}\left(\frac{\lambda}{e}\right)^{\lambda}}{\prod_{k=1}^K\sqrt{\lambda p_k}\left(\frac{\lambda p_k}{e}\right)^{\lambda p_k}}\cdot\prod_{k=1}^K (q_k)^{\lambda p_k} \\ &\sim \lambda^{\frac{K-1}{2}}\cdot\prod_{k=1}^K\left(\frac{q_k}{p_k}\right)^{\lambda p_k}.\end{aligned} \tag{B.6}$$

Write $g(\boldsymbol{q}) = \prod_{k=1}^K\left(\frac{q_k}{p_k}\right)^{p_k}$, we now get the maximization of $g(\boldsymbol{q})$ w.r.t. $\boldsymbol{q}$. Since we have the constrain that $\sum_{k=1}^K q_k = 1$, we use the Lagrange multiplier to find the maximum of $g(\boldsymbol{q})$. The Lagrangian function is defined as

$$\mathcal{L}(q_1,\ldots,q_K,c) \equiv \log g + c\left(1-\sum_{k=1}^K q_k\right), \tag{B.7}$$

where $\log g = \sum_{k=1}^K p_k(\log q_k - \log p_k)$.

To solve

$$\nabla_{q_1,\cdots,q_K,c} g(q_1,\cdots,q_K,c) = 0, \tag{B.8}$$

we have

$$\begin{cases}\frac{p_k}{q_k} - c = 0, & \text{for } k=1,\ldots,K \\ \sum_{k=1}^K q_k = 1\end{cases} \tag{B.9}$$

since $\sum_{k=1}^K p_k = 1$, we can easily get that the solution is

$$\begin{cases} q_k = p_k, & \text{for } k=1,\cdots,K \\ c=1.\end{cases} \tag{B.10}$$

That is, we have $g_{\max}(\boldsymbol{q})=1$ when (B.10) holds, and this leads to $\pi_{\boldsymbol{q}}(\boldsymbol{q};\boldsymbol{p},\lambda) \sim \lambda^{\frac{K-1}{2}}\cdot 1^{\lambda} \overset{\lambda\to\infty}{\longrightarrow} \infty$.

For $\forall \boldsymbol{q} \neq \boldsymbol{p}$, we have $g(\boldsymbol{q}) < 1$, which leads to $\pi_{\boldsymbol{q}}(\boldsymbol{q};\boldsymbol{p},\lambda) \sim \lambda^{\frac{K-1}{2}}\cdot(g(\boldsymbol{q}))^{\lambda} \overset{\lambda\to\infty}{\longrightarrow} 0$. Thus, as $\lambda\to\infty$, we have

$$\begin{aligned}&\lim_{\lambda\to\infty}\int \pi_{\boldsymbol{q}}(\boldsymbol{q};\boldsymbol{p},\lambda)\mathrm{d}\boldsymbol{q} = 1, \\ &\pi_{\boldsymbol{q}}(\boldsymbol{q};\boldsymbol{p},\lambda) = 0 \text{ if } \boldsymbol{q}-\boldsymbol{p} \neq 0.\end{aligned} \tag{B.11}$$

That is, $\pi_{\boldsymbol{q}}(\boldsymbol{q};\boldsymbol{p},\lambda) \longrightarrow \delta(\boldsymbol{q}-\boldsymbol{p})$ as $\lambda\to\infty$.

If $\exists\, k$ s.t. λp_k is not an integer. According to the Stirling formula for the gamma function that $\Gamma(z) \sim \sqrt{\frac{2\pi}{z}}\left(\frac{z}{e}\right)^z$, following the same analysis, we can also get $\pi_{\boldsymbol{q}}(\boldsymbol{q};\boldsymbol{p},\lambda) \longrightarrow \delta(\boldsymbol{q}-\boldsymbol{p})$ as $\lambda\to\infty$.

□

C. Simulation

We conduct simulation studies to evaluate the performance of the proposed BKD method in classification problems. We consider four distinct scenarios with varying dimensions and number of classes. To evaluate model performance, we employ two criteria: classification accuracy and mean absolute error (MAE) on class probability p. In addition to assessing the model's overall accuracy, we place particular emphasis on validating its capacity to quantify uncertainty. To achieve this, we use the mean deviance as a metric to measure prediction uncertainty and explore its relationship with the original data. Additionally, we analyze coverage rates across various confidence levels and sample sizes, providing a comprehensive assessment of the method's uncertainty quantification ability.

C.1. Simulation Settings

We consider four different scenarios for generating synthetic data, dividing the data into training, validation, and testing sets in a 7:3:1 ratio for all scenarios.

In the context of binary classification problems, we explore three scenarios involving different dimensions and model structures.

$$\begin{aligned} p(\mathbf{x}) &= \frac{\exp(\eta(\mathbf{x}))}{1+\exp(\eta(\mathbf{x}))} \\ Y &\sim Bernoulli(p(\mathbf{x})) \end{aligned} \tag{C.1}$$

Scenario 1: $\eta(\mathbf{x}) = 2 - 2x_1 + x_2$,

where $x_1 \sim Unif(-4, 6)$ and $x_2 \sim Unif(-4, 4)$. We generate a sample of size $10,000$.

Scenario 2: $\eta(\mathbf{x}) = 1 - 2x_1 + x_2 - x_3 - 0.5x_4 + 2x_5$,

where $x_1, x_5 \sim Unif(-2, 4)$, $x_2, x_3 \sim Unif(-4, 4)$, and $x_4 \sim Unif(0, 2)$. We generate a sample with sample size $10,000$.

Scenario 3: $\eta(\mathbf{x}) = 2\exp(x_1) + \frac{1}{2}x_2^2 + 5sin(x_3x_4) + \frac{1}{2}\sum_{j=5}^{10} x_j - 3$,

where $\mathbf{x} = (x_1, \ldots, x_k)^T \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with k-dimensional mean vector $\mu_i = 0$ and $k \times k$ covariance matrix $\Sigma_{i,j} = \rho^{|i-j|}$, with $1 \le i \le d$ and $1 \le j \le d$. We set $d = 20$, $\rho = 0.8$, and generate a sample with sample size $20,000$.

Scenario 4: We also consider a multi-class scenario, using the model described below. We set

$$\begin{aligned} p_i(\mathbf{x}) &= \frac{f(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma})}{\sum_{j=1}^{5} f(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})},\ i = 1, \ldots, 5 \\ \boldsymbol{p}(\mathbf{x}) &= (p_1(\mathbf{x}), p_2(\mathbf{x}), p_3(\mathbf{x}), p_4(\mathbf{x}), p_5(\mathbf{x}))^T, \\ Y &\sim Multinomial(\boldsymbol{p}(\mathbf{x})) \end{aligned} \tag{C.2}$$

where $\boldsymbol{\mu}_1 = (0, 0, 0, 0, 0)^T$, $\boldsymbol{\mu}_2 = (3, 3, 3, 3, 3)^T$, $\boldsymbol{\mu}_3 = (0, 0, 1, 3, 2)^T$, $\boldsymbol{\mu}_4 = (2, 0, 1, 2, 1)^T$, $\boldsymbol{\mu}_5 = (2, 2, 1, 0, 1)^T$, and $\Sigma = (0.5^{|i-j|})_{5\times5}$. We simulate $8,000$ sample from each $N(\boldsymbol{\mu}_i, \Sigma)$, $i = 1, \ldots, 5$, and generate labels according to Equation (C.2).

C.2. Details on Model Structure

Scenario 1: The teacher model employs a Multilayer Perceptron (MLP) architecture consisting of five hidden layers. These layers have 7, 10, 12, 10, and 5 nodes respectively. The model uses the ReLU activation function and incorporates a dropout rate of 0.2. The student model employs an MLP architecture consisting of one hidden layer with five nodes. The model uses the ReLU activation function.

Scenario 2: The teacher model employs a Multilayer Perceptron (MLP) architecture consisting of four hidden layers. These layers have 5, 8, 12, and 5 nodes respectively. The model uses the ReLU activation function and incorporates a dropout rate of 0.1. The student model employs an MLP architecture consisting of one hidden layer with five nodes. The model uses the ReLU activation function.

Scenario 3: The teacher model employs a Multilayer Perceptron (MLP) architecture consisting of five hidden layers. These layers have 30, 20, 10, 10, and 5 nodes respectively. The model uses the ReLU activation function and incorporates a dropout rate of 0.2. The student model employs an MLP architecture consisting of two hidden layers. These layers have 25, and 10 nodes respectively. The model uses the ReLU activation function.

Scenario 4: The teacher model employs a Multilayer Perceptron (MLP) architecture consisting of five hidden layers. These layers have 7, 15, 12, 10, and 7 nodes respectively. The model uses the ReLU activation function and incorporates a dropout rate of 0.2. The student model employs an MLP architecture consisting of one hidden layer with ten nodes. The model uses the ReLU activation function.

C.3. Classification Results

We evaluate the classification results using both accuracy and MAE. MAE measures the difference between the estimated class probability and the true class probability p. For the proposed BKD method, MAE is calculated by comparing the predicted posterior mode with the true class probability p. The results are detailed in Table 4. We compare BKD with the teacher model and the original KD method. The best results in each scenario are highlighted in bold.

	Accuracy			MAE		
Dataset	Teacher	Orig KD	BKD	Teacher	Orig KD	BKD
Scenario 1	**0.921**	0.918	0.920	0.069	0.027	**0.023**
Scenario 2	0.897	0.896	**0.901**	0.090	0.088	**0.028**
Scenario 3	**0.868**	0.858	**0.868**	0.112	0.122	**0.089**
Scenario 4	0.838	0.862	**0.870**	0.076	0.044	**0.037**

Table 4. Accuracy and MAE for different methods in simulation scenarios. The best results for each scenario are marked in bold.

The results in Table 4 highlight the efficiency of the BKD method. Notably, BKD achieves the highest accuracy in three out of four scenarios and consistently records the lowest MAE, demonstrating its robust performance across different conditions. Specifically, in Scenario 2, BKD improves accuracy from 0.897 (teacher model) to 0.901, and in Scenario 4, it shows a more notable increase from 0.838 to 0.870, indicating its strength in enhancing the model's classification performance. Moreover, BKD considerably reduces the MAE, to about half that of the other methods, demonstrating its precision in accurately estimating class probabilities in addition to class labels. Notably, in Scenario 1, while the teacher model exhibits the highest accuracy at 0.921, BKD is close behind at 0.920, along with a substantial MAE reduction from 0.069 to 0.023. A similar trend is observed in Scenario 3. These slight differences in accuracy, paired with notable decreases in MAE, indicate that the advantages of BKD extend beyond mere accuracy. It can effectively predict class probabilities, thereby contributing to the model's reliability and enhancing its generalization capacity.

C.4. Uncertainty Evaluation

C.4.1. Mean deviance

We employ mean deviance as a metric to quantify model uncertainty at each data point. In our model, where $Y \sim Bernoulli(p)$, the variance of each observation is highly correlated with $p(1-p)$. Therefore, the class probability p is a critical determinant of the prediction uncertainty for each data point. In Figure 9, this relationship is illustrated by plotting mean deviance against various values of p across the first three scenarios. We observe that the general trend is that as p increases, the mean deviance first increases and then decreases. Data points with p closer to 0.5 are associated with higher mean deviance, indicating greater prediction uncertainty. This observation aligns with theoretical expectations, as predictions on data points with probabilities near 0.5 are inherently more uncertain.

Despite the overall trend of uncertainty correlating with class probability p, as described above, there is noticeable variation in the uncertainty of data points that share the same class probability p. This variation is especially pronounced in Scenario 3. This phenomenon suggests the presence of additional factors, beyond a data point's class probability, contributing to the model's predictive uncertainty. To investigate this, we apply t-SNE (Van der Maaten & Hinton, 2008) for dimensional reduction of data points with class probabilities p near 0 or 1 into a two-dimensional space. The resulting visualization, shown in Figure 10, color-codes data points according to their class probability p in the left panel and mean deviance in the right panel. We observe that although all data points highlighted in yellow share a similar class probability p, those in the top

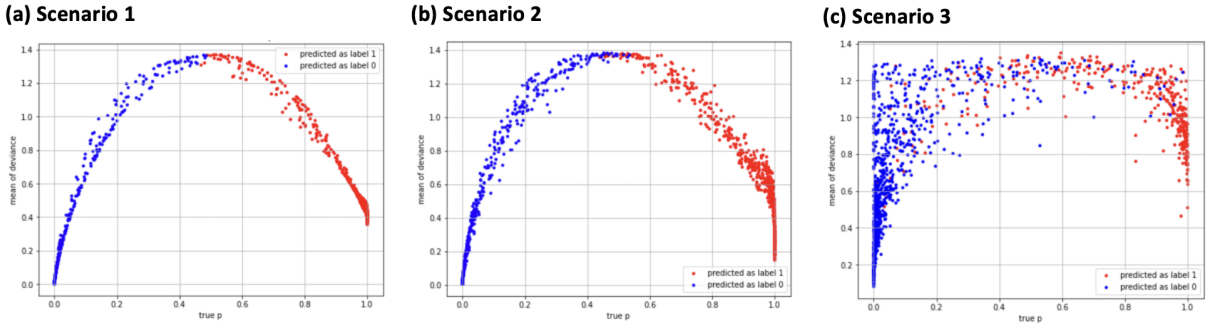

Figure 9. Relationship between the mean deviance and true class probability p for simulation scenarios 1-3.

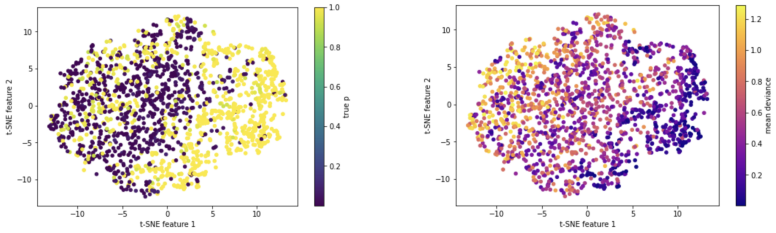

Figure 10. Two-Dimensional Visualization of Simulated Data in Scenario 3 Using t-SNE. In the left panel, data points are colored based on class probability p. In the right panel, data points are colored based on the mean deviance calculated using our BKD method. The results demonstrate that mean deviance is influenced not only by p but also by additional factors, such as neighborhood information.

left corner exhibit lower variance. Since these data points are predominantly surrounded by others with comparable values of p, it is plausible that the model considers neighboring data points when making predictions. The model tends to be more certain of its predictions when the surrounding data points strongly resemble the data point in question. In contrast, data points in the lower right corner are surrounded by numerous points from different classes. Consequently, the model tends to produce predictions with higher uncertainty for these data points. This observation suggests that the variations in prediction uncertainty are influenced by neighboring data points. This phenomenon is consistent with the toy example in the main text.

C.4.2. Coverage Rate

We calculate the coverage rate using Equation (17) and conduct tests at three commonly used credible levels (0.85, 0.90, 0.95), with varying proportions of testing data (0.05, 0.1, 0.2, 0.4, 0.6, 0.8, 1.0). Each test setting is replicated 10 times. The results, presented in Figure 11, reveal that our BKD method consistently achieves coverage rates close to the specified credible levels. As the size of the testing sample increases, variation across repetitions decreases, and the mean coverage rate stabilizes, closely aligning with the chosen credible level.

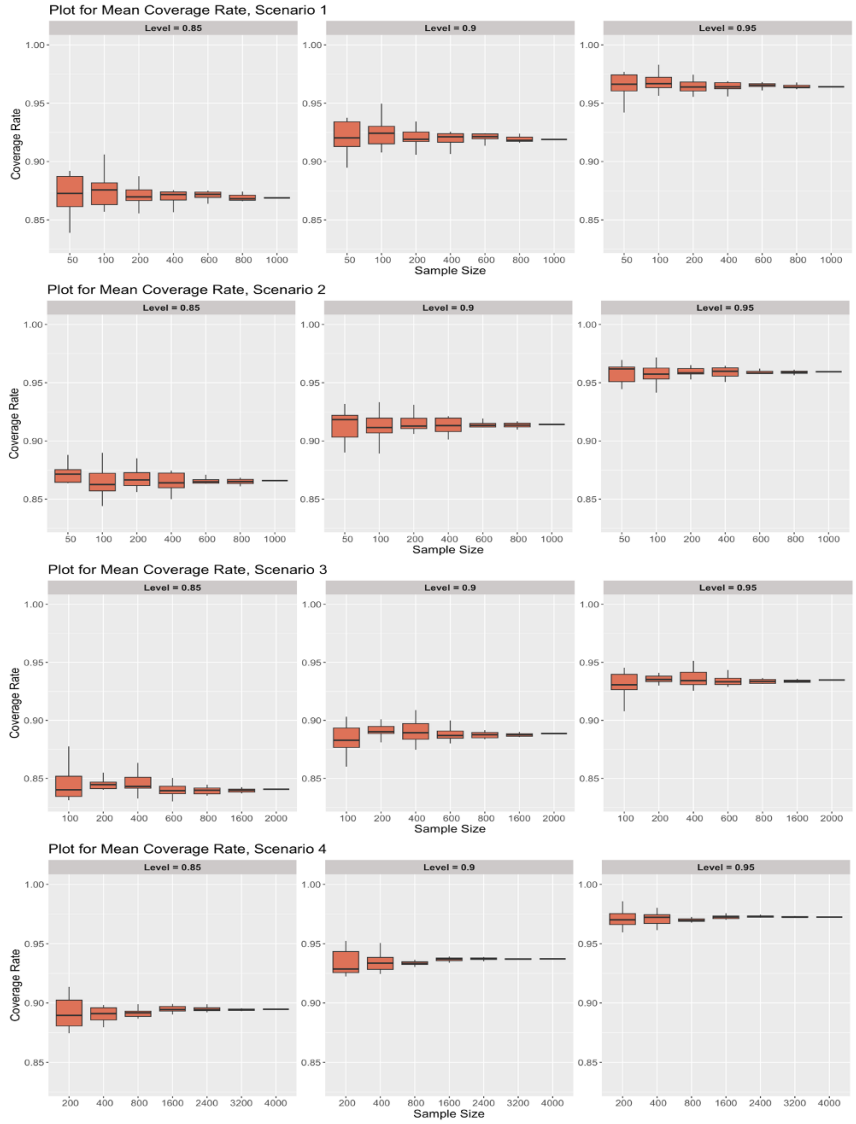

Figure 11. Coverage rate across commonly used credible levels in simulation scenarios.

C.5. BKD with the teacher model as QDA and the Student model as LDA

In the case where we consider the student model to be LDA, we can derive the posterior distribution of the class means and the common covariance matrix across all classes as a normal-inverse-Wishart distribution. This allows for analytical Bayesian inference.

If the student model is a classifier by linear discriminant analysis (LDA), we have parameter $\boldsymbol{\theta} = \{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K, \rho_1, \cdots, \rho_K, \boldsymbol{\Sigma}\}$, where $\boldsymbol{\mu}_k$ is the kth class mean, ρ_k is the prior probability of being kth class, $\boldsymbol{\Sigma}$ is the common covariance matrix K classes.

We can derive

$$\log(h_k(\mathbf{x}_i, \boldsymbol{\theta})) = -\|(\mathbf{x}_i - \boldsymbol{\mu}_k)\boldsymbol{\Sigma}^{-1/2}\|_2^2/2 - \log(|\boldsymbol{\Sigma}|)/2 + \log \rho_k + C,$$

where C is a constant. Through this, the posterior distribution of $\boldsymbol{\Sigma}$ is the inverse Wishart distribution $\mathcal{W}^{-1}(\mathbf{S}, n(1+\lambda))$, and

$$\mathbf{S} = \sum_{k=1}^{K}\sum_{i=1}^{n}(y_{ik} + \lambda p_{ik})\mathbf{x}_i^T\mathbf{x}_i - \sum_{k=1}^{K} M_k \bar{\boldsymbol{\mu}}_k^T \bar{\boldsymbol{\mu}}_k,$$

where $M_k = \sum_{i=1}^{n}(y_{ik} + \lambda p_{ik})$, $\bar{\boldsymbol{\mu}}_k = \sum_{i=1}^{n}(y_{ik} + \lambda p_{ik})\mathbf{x}_i / M_k$. What's more, we have the posterior distribution of $\boldsymbol{\mu}_k$ conditioned on $\boldsymbol{\Sigma}$ is the multivariate normal distribution $MVN(\bar{\boldsymbol{\mu}}_k, \Sigma/M_k)$.

To validate this approach, we conduct experiments where the teacher model is trained with QDA and the student model is LDA. Specifically, we generate synthetic data similar to simulation scenario 4 by setting

$$\begin{aligned} Pr(Y = k|\mathbf{x}) &= \frac{Pr(\mathbf{x}|Y = k)Pr(Y = k)}{\sum_{l=1}^{2} Pr(\mathbf{x}|Y = l)Pr(Y = l)}, \\ Pr(\mathbf{x}|Y = k) &= f(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma_k}) \end{aligned} \tag{C.3}$$

where $\boldsymbol{\mu}_1 = (0, 0, 0, 0)^T$, $\boldsymbol{\mu}_2 = (0, 0, 3/2, 2/3)^T$, $\boldsymbol{\Sigma}_1 = (0.7^{|i-j|})_{4\times4}$, and $\boldsymbol{\Sigma}_2 = (0.4^{|i-j|})_{4\times4}$. We report (1) accuracy and mean absolute error (MAE) of the estimated probability (Table 5); (2) mean coverage rate of the credible interval (Table 6); and (3) the relationship between mean deviance $\overline{\Delta}$ (uncertainty) and true probability $p \triangleq Pr(Y = 1|\mathbf{x})$.

Table 5. Accuracy and MAE of QDA, LDA, and BKD on Simulated data.

	QDA (teacher)	LDA (student)	BKD (our)
Accuracy	0.845	0.837	0.842
MAE	0.064	0.144	0.138

Table 6. Coverage rate of BKD with varying sample sizes, with standard deviation in parenthesis. Credible level is set to be 0.95.

	n=480	n=960	n=1440	n=1920
Level = 0.95	0.9504	0.9484	0.9477	0.9473
	(0.0060)	(0.0038)	(0.0023)	(0.0011)

In Figure 12, we plot the mean deviance against various values of p. The phenomenon is similar to the simulation settings. We observe that the general trend is that as p increases, the mean deviance first increases and then decreases. Data points with p closer to 0.5 are associated with higher mean deviance, indicating greater prediction uncertainty. This observation aligns with expectations, as predictions on data points with probabilities near 0.5 are inherently more uncertain. All results show that in this scenario, BKD also successfully distills knowledge from the teacher model.

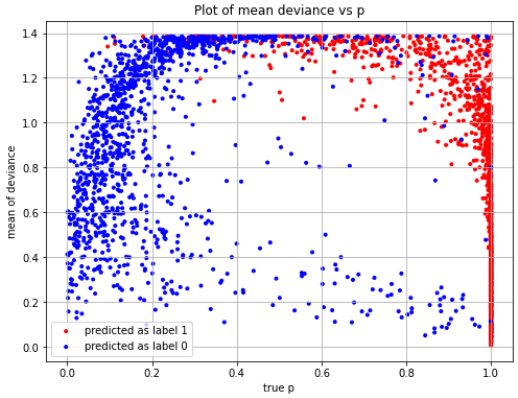

Figure 12. Relationship between the mean deviance and true class probability p.

D. Real Data Analysis

D.1. Details on Data Sets

MNIST: MNIST (LeCun, 1998) is a dataset of handwritten digit images with a training set of $60,000$ examples and a test set of $10,000$ examples. Each example is a 28×28 grayscale image, associated with a label of 10 classes.

Fashion MNIST: Fashion MNIST (Xiao et al., 2017) is a dataset of Zalando's article images with a training set of $60,000$ examples and a test set of $10,000$ examples. Each example is a 28×28 grayscale image, associated with a label from 10 classes.

CIFAR-10: The CIFAR-10 dataset (Krizhevsky et al., 2009) consists of a training set of $50,000$ examples and a test set of $10,000$ examples. Each example in the dataset is a 32×32 color image, spanning 10 different classes of objects such as animals and vehicles. These classes include airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks, each equally represented in the dataset.

CIFAR-100: CIFAR-100 (Krizhevsky et al., 2009) is an image dataset that consists of a training set of $50,000$ examples and a test set of $10,000$ examples. Each example is a 32×32 color image, categorized into 100 classes.

D.2. Details on Model Structure

MNIST dataset: The model structure is inspired by Hinton et al. (2015). Specifically, the teacher model employs an MLP with two hidden layers of 1200 hidden nodes. The model uses the ReLU activation function and incorporates a dropout rate of 0.5. The model also incorporates a dropout layer with rate 0.2 for the input. The student model employs an MLP architecture consisting of two hidden layers. These layers have 200 and 100 nodes, respectively. The model uses the ReLU activation function.

Fashion MNIST dataset: The teacher model employs the 50-layer ResNet architecture as described by He et al. (2016). We adhered closely to the parameter settings of this model as implemented in the PyTorch torchvision library (Paszke et al., 2019). The ResNet50 model has 25.6 million parameters. The student model employs a CNN architecture with two convolutional layers and a subsequent fully connected layer. The first convolutional layer has 16 output channels, batch normalization, ReLU activation, and max pooling. The second convolutional layer follows a similar pattern but with 8 output channels. The convolutional layer is then followed by fully connected layers, consisting of a hidden layer with 60 nodes and an output layer with 10 nodes.

CIFAR-10 / CIFAR-100 datasets: The teacher model employs the ViT-B-16 architecture as described by Dosovitskiy et al. (2020). The ViT-B-16 model has 86 million parameters. The student model uses the MUXNet-m architecture proposed in Lu et al. (2020), with around 3.4 million parameters.

D.3. Implementation Details of BNN and Dropout

To implement the method integrating the original KD and BNN, we set the BNN model as the student. We train this model using variational inference with details described in Shridhar et al. (2019), and incorporate the KD loss to distill knowledge from the teacher model. An important hyperparameter when training the BNN model is the weight of the KL loss. We perform a grid search over the values $\{0.01, 0.05, 0.1, 0.2\}$ to find the parameter that yields the highest accuracy. For the CIFAR-10 and CIFAR-100 datasets, due to computational constraints, we add the Bayesian layers only after the feature extraction blocks.

To implement the method integrating the original KD and Monte Carlo dropout, we enable dropout during the testing/inference stage to obtain multiple stochastic predictions.

D.4. BKD on the MNIST dataset using continuous categorical distribution (CC)

We notice that Gordon-Rodriguez et al. (2020) propose the continuous categorical distribution given the concern of Dirichlet distribution's divergence for modeling the probability vectors close to the extrema of the simplex. Therefore, changing the prior to continuous categorical distribution could be beneficial for the tasks when the student model can reach very high prediction accuracy. In particular, we consider a continuous categorical distribution $\mathcal{CC}(1 + \lambda \mathbf{p}_i)$, i.e.,

$$\pi_{\mathbf{q}}(\mathbf{q}, \mathbf{p}_i) \propto \Pi_{k=1}^{K} (1 + \lambda p_{ik})^{q_k},$$

for $\lambda > 0$. Note that as $\lambda \to \infty$, we also have $\pi_{\mathbf{q}}(\mathbf{q}, \mathbf{p}_i) \to \delta(\mathbf{q} - \mathbf{p}_i)$ as described in Theorem 4.2.

We evaluate the performance of BKD on the MNIST dataset using Dirichlet prior (Dir) and continuous categorical distribution (CC), independently. Both priors yield comparably high model accuracies, with Dir 0.980 and CC 0.979. Table 7 details mean deviance (uncertainty) when taking the perturbed images with different perturbation levels γ as input. BKD's mean deviance with the Dirichlet prior consistently rises with increased perturbation (γ), while it initially increases and then

slightly fluctuates with the CC prior. Nevertheless, the fluctuations in CC are minimal and remain within a reasonable range. In contrast, the original KD method tends to yield overly confident predictions at high perturbation levels. These findings demonstrate BKD's robustness to prior selection, indicating that various reasonable priors can produce similarly robust and satisfactory outcomes.

Table 7. Mean deviance (uncertainty) with varying perturbation levels, on the MNIST dataset.

	γ=0	γ=0.1	γ=0.2	γ=0.3	γ=0.4	γ=0.5	γ=0.6	γ=0.7	γ=0.8	γ=0.9	γ=1
Orig KD	0.03	0.07	0.33	0.74	**0.80**	0.77	0.76	0.68	0.58	0.47	**0.38**
BKD (Dir)	0.08	0.15	0.41	0.74	1.19	1.39	1.42	1.42	1.41	1.43	**1.46**
BKD (CC)	0.16	0.31	0.91	1.63	2.24	2.73	2.97	**2.99**	2.93	2.90	**2.92**

D.5. More Results

We present a separate visualization for each class within the Fashion MNIST dataset in Figure 13.

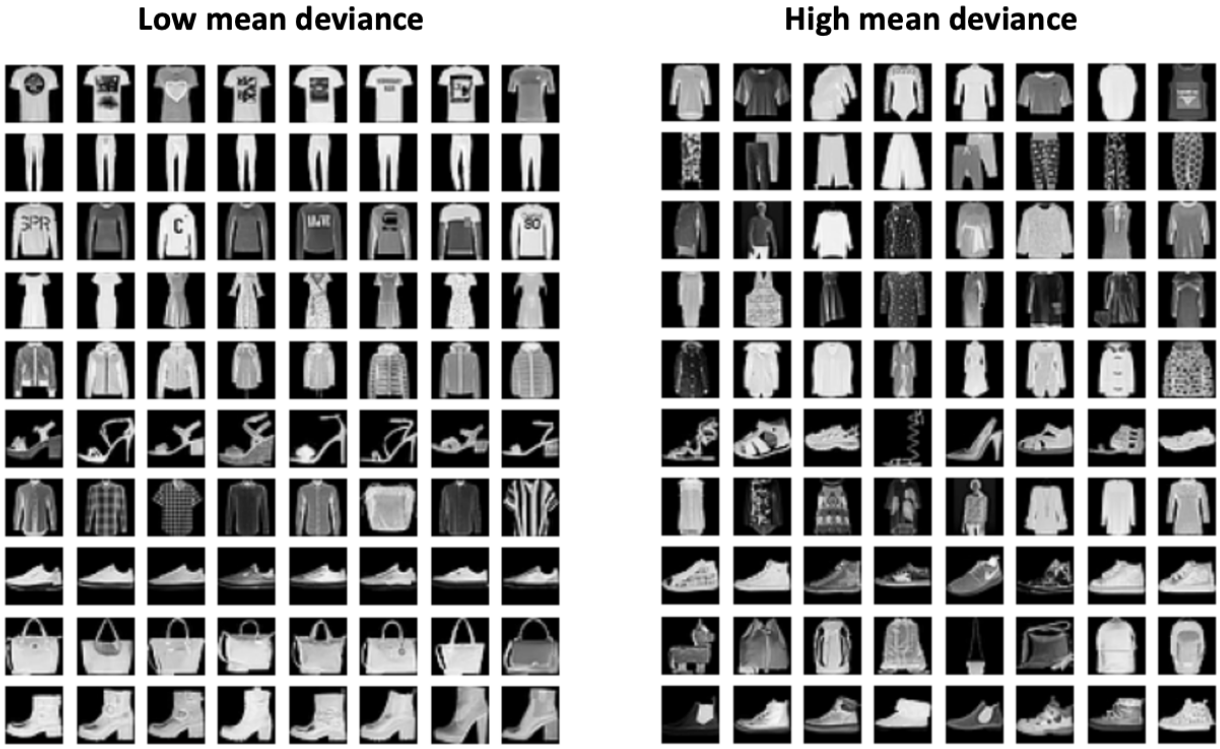

Figure 13. For each class in the Fashion MNIST dataset, the 8 images with the lowest mean deviance are displayed on the left, while the 8 images with the highest mean deviance are shown on the right.

The distribution of mean deviance for the MNIST dataset is shown in Figure 14. The first box displays the log-transformed mean deviance across all images, while the subsequent boxes individually represent the log-transformed mean deviance for images within each class. The results show that, on average, digits '0' and '6' have lower mean deviance, indicating higher prediction confidence for these digits. Conversely, digits '8' and '9' exhibit higher mean deviance. This observation aligns with our intuition as digit '0' is marked by its unique round shape and digit '6' is distinguishable due to its notable loop at the bottom. On the contrary, digit '8' can often be confused with '3' when its top loop is small or open, and it can also be mistaken for a '9' if the bottom loop is written more openly. While a loosely written '9', with its loop not fully closed, can resemble '4'. Additionally, a '9' with a long stem and tiny loop could be misidentified as '7', particularly when '7' is written with a crossbar.

The distribution of mean deviance for the CIFAR-10 dataset is shown in Figure 15. We find that on average cars are

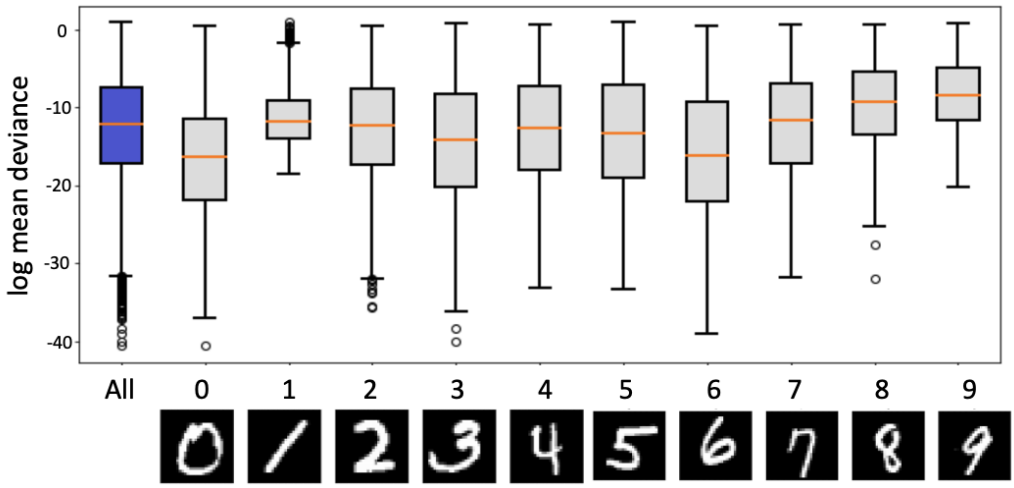

Figure 14. The boxplot illustrates the log-transformed mean deviance for the MNIST dataset. The leftmost box shows the result for all images, whereas the following boxes detail the results for each class.

the simplest to classify, with ships coming next. This aligns with our expectations, as the shape of cars is simple and characterized by recognizable elements such as wheels and windows. Ships, distinct in both large size and distinct shape among the ten categories, also facilitate easier classification. The most challenging classes to classify are cats and dogs, with cats posing a slightly greater challenge. The primary difficulty arises from the high variability within these classes, including a wide range of breeds, various poses, and their often complex surroundings. Furthermore, the similarity in features shared between cats and dogs, such as their fur and comparable body structures, adds complexity to their accurate classification.

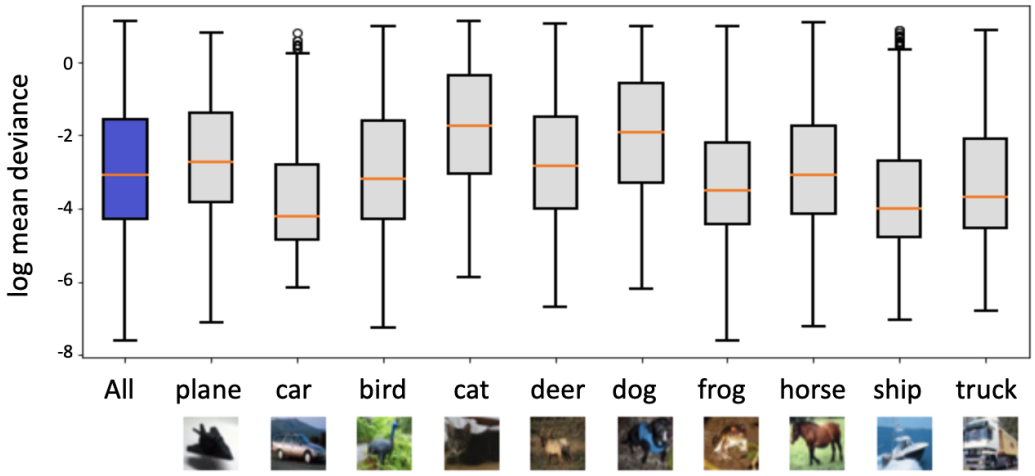

Figure 15. The boxplot illustrates the log-transformed mean deviance for the CIFAR-10 dataset. The leftmost box shows the result for all images, whereas the following boxes detail the results for each class.

For the CIFAR-100 dataset, due to the large number of categories (100), we omit the results of additional analyses here.

The coverage rate is shown in Figure 16. In some cases, the results from BNN and Dropout are highly consistent across repetitions, resulting in box plots that appear as straight lines. For instance, with the MNIST dataset, BNN consistently achieves a coverage rate of around 0.86, whereas Dropout reaches a coverage rate equal to 1. In the case of the Fashion MNIST dataset, BNN attains a coverage rate of approximately 1, while Dropout often has a coverage rate of around 0.85. For the CIFAR-10 dataset, BNN consistently shows a coverage rate equal to 1.

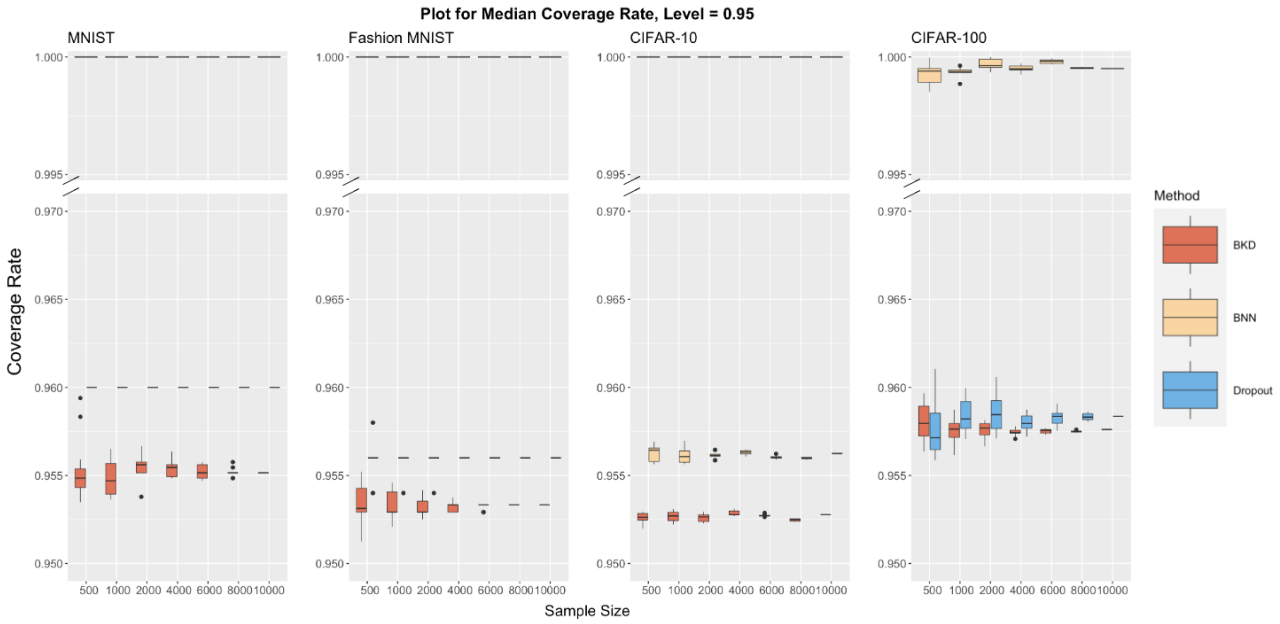

Figure 16. Evaluation of coverage rate on commonly used credible level 0.95 for various sample sizes in MNIST, Fashion MNIST, CIFAR-10, and CIFAR-100 datasets. We compare the results of BKD, BNN, and Dropout.