Uncertainty-guided Curriculum Learning via Infinitesimal Jackknife

Zifeng Wang[1], **Rui Wen**[2], **Xi Chen**[2], **Ercan E. Kuruoglu**[1,3], **Shao-Lun Huang**[1], **Yefeng Zheng**[2]
[1]Tsinghua-Berkeley Shenzhen Institute, Tsinghua University [2]Jarvis Lab, Tencent [3]On leave from ISTI-CNR
Email: wangzf18@mails.tsinghua.edu.cn

Abstract

One of the key challenges lies in curriculum learning (CL) is how to design reasonable metrics for scoring sample difficulty. Current data-driven CL algorithms need learning extra parameters for sample scoring task. In contrast, we propose uncertainty-guided curriculum learning (UCL), where the sample difficulty is scored according to both the uncertainty in model's decision-making (epistemic) and the uncertainty inherent in data (aleatoric). We also show a novel scoring metric based on signal-to-noise ratio (SNR), which combines the uncertainty and magnitude of prediction together to evaluate sample difficulty synthetically. Moreover, in order to avoid burdensome learning on Bayesian neural network, we propose to use infinitesimal jackknife (IJ) to quantify uncertainty in model weights through a pseudo ensemble approach based on Fisher information matrix. Experiments on various image and text datasets empirically verify the superiority of UCL over the state-of-the-art CL baselines. The source code of our algorithm is available in supplementary materials.

1 Introduction

Curriculum learning (CL) (Bengio et al. 2009) garners attention recently because of its efficacy in boosting convergence as well as generalization ability of machine learning models, by rearranging sampling orders of training data. In particular, CL proposes to utilize simple samples prior to hard samples at the initial stage of training, thereafter gradually level up the ratio of hard samples until convergence. This learning formula is similar to how humans receive a series of organized curriculum in education system, thus acquiring elementary to advanced knowledge stage by stage. This idea was initially studied in animal training where it is called *shaping* (Skinner 1958; Peterson 2004) and not introduced to machine learning until 2009 (Bengio et al. 2009).

Even though the predefined curriculum by humans has been proved to improve model performance in a variety of tasks, there still lie open problems that probably render CL difficult to implementation and useless sometimes. In this circumstance, we firstly need to deal with the semantic ambiguity on the *difficulty* of samples, namely *scoring* (Hacohen and Weinshall 2019). Traditional wisdom (Bengio et al. 2009; Jiang et al. 2015) proposed to adopt domain knowledge as a priori to difficulty ranking. Nonetheless, it is argued that the manually designed CL is costly, and human teachers are unable to evaluate the true difficulty as it affects the student model. It typically emerges with adversarial examples (Szegedy et al. 2013) which are confusing to neural networks but easy to human beings. Moreover, the fixed curriculum is unnecessarily optimal due to its oversight of feedback from the student model (Jiang et al. 2018).

Considering above challenges, a surge of research advocates to discover data-driven curriculums from data, including MentorNet (Jiang et al. 2018), CurriculumNet (Guo et al. 2018), Transfer CL (Weinshall, Cohen, and Amir 2018), etc. These data-driven curriculums have demonstrated superiority than the predefined curriculum. However, these methods often require burdensome steps to learn a powerful teacher model that is responsible for sorting the training data at hand. More importantly, few of them dig into the *uncertainty* in model decision making process, that is, how confident the model is when makes prediction on each sample. Kendall and Gal (2017) showed that the Bayesian neural network (BNN) is often more uncertain on samples which are rare in training set as well as on those are blurred. In this work, we value quantifying predictive uncertainty of models because it is a natural metric for sample difficulty but not yet well explored in the literature of CL. Besides, this uncertainty-guided approach sheds light on interpretability in CL, i.e., we can be aware of why one sample is regarded difficult by models based on the quantifiable uncertainty measurement.

One would notice that there appear attempts to incorporate BNN into CL application to neural machine translation by Zhou et al. (2020) recently. They follow the classical Monte Carlo (MC) dropout (Gal and Ghahramani 2016) paradigm to approximate Bayesian inference. However, MC dropout can only capture the model uncertainty, i.e., **epistemic uncertainty**, but cannot capture noise inherent in the input data, i.e., **aleatoric uncertainty**. Hence they employ a language model, e.g., BERT (Devlin et al. 2019), to infer the aleotric uncertainty during training. The involvement of BERT in training incurs the concern of extra time consumption that might nullify the gain in the speeding up by CL.

In this work, on contrast, we utilize a unified Bayesian framework that combines two uncertainties together in a light-weight manner. Besides, instead of modeling weight distribution as isotropic Gaussian as many previous BNN works did (Gal and Ghahramani 2015; Blundell et al. 2015) for the sake of optimization tractability, here we propose to use a novel pseudo ensemble approach based on infinitesimal jackknife (IJ) estimation (Jaeckel 1972). Our approach enables modeling full covariance matrix of the weight distribution without optimization, thus enhancing the accuracy of

Preliminary Work.

uncertainty inference. With a paralleled MC sampling implementation, our approach can infer two uncertainties at the same time with very low expense. Based on the predictive uncertainty, we further design four scoring methods: epistemic, aleatoric, total and signal-noise-ratio (SNR). Our main contributions are listed as following:

- We introduce a novel uncertainty-guided curriculum learning (UCL) method that can infer both aleatoric and epistemic uncertainties at the same time to obtain the multifaceted sample difficulty.
- We define a novel uncertainty transfer method, along with a novel scoring metric, namely signal-to-noise ratio (SNR), which combines the uncertainty and magnitude of prediction together to decide sample difficulty.
- We employ a fresh infinitesimal jackknife (IJ) approach to realize variational inference with full covariance estimation with low computational cost.
- Our CL method firstly covers both image and text data, and proves to acquire promising results empirically in many open datasets.

2 Related Work

2.1 Curriculum Learning

Curriculum learning emerged as an active research area in machine learning community pioneered by Bengio et al. (2009), where the curriculum is predefined by human teachers and keeps fixed during the course of optimization. Later, Kumar, Packer, and Koller (2010) proposed a self-paced learning (SPL) approach that optimizes model parameters as well as curriculum at the same time, by assigning an additional trainable weight variable for each sample. Each weight is compared with an gradually increasing threshold λ that decides whether the corresponding sample is picked during training. SPL was then followed by a series variants, e.g., self-paced curriculum learning (Jiang et al. 2015), self-paced learning with implicit regularization (Fan et al. 2017) and data parameters (Saxena, Tuzel, and DeCoste 2019). However, they scale the volume of sample weights linearly with the sample size and require tremendous efforts in tuning hyperparameters about step size of λ, penalty and learning rate on weight variables.

Recent works emphasize on utilizing another powerful teacher model to guide the student model to learn, which is engaged with unsupervised learning in Guo et al. (2018), meta-learning in Jiang et al. (2018), transfer learning in Weinshall, Cohen, and Amir (2018), knowledge distillation in Dogan et al. (2019), etc. These approaches involve a pre-training step on a teacher model at a task, and then use the teacher model to generate sample weights for student model training at target task. On contrast, we value a simple yet effective method that does not require held-out dataset, and provides interpretable curriculum that is consistent with model's "awareness" of sample difficulty.

2.2 Bayesian Deep Learning

Accompanied with the brilliant success of deep learning since 2012 Krizhevsky, Sutskever, and Hinton (2012), Bayesian deep learning also thrives as the Bayesian counterpart of the frequentist deep learning. Before that, several approaches have been proposed for Bayesian neural networks (BNNs) based on, e.g., the Laplace approximation (MacKay 1992), Hamiltonian Monte Carlo (Neal 2012), variational inference (Hinton and Van Camp 1993; Graves 2011), probabilistic backpropagation (Hernández-Lobato and Adams 2015), MC dropout (Gal and Ghahramani 2016), etc. Armed with Bayesian deep learning, uncertainty-guided methods succeed in the realm of machine learning, including active learning (Gal, Islam, and Ghahramani 2017), noise label learning (Wang, Kucukelbir, and Blei 2017), continual learning (Nguyen et al. 2018), etc. Predictive uncertainty in BNN is also a natural metric for sample difficulty, as it indicates how confidence when a model makes prediction on input samples. Recently, Zhou et al. (2020) pioneered in uncertainty-aware CL, while their method is specifically for neural machine translation thus needing adjustment for other applications. In this work, we proposed a novel UCL framework that can capture both epistemic and aleatoric in an end-to-end way, which is efficient and can scale to a broad scope.

3 Main Method

In this section, we present main techniques of the proposed UCL framework. As aforementioned, the key idea of UCL is to measure sample difficulty in a Bayesian paradigm. To this end, we first present the basic conception about BNN in contrast to deterministic neural network (DNN), and the classical practice in BNN. Then, we propose a novel IJ-based method to quantify uncertainty in BNN, which is in fact a *pseudo ensemble* of infinite DNN models. Finally, we describe how to acquire epistemic and aleatoric uncertainty in single inference, without introduction of extra trainable variance terms.

3.1 Uncertainty in Bayesian Deep Learning

Given a training set $\mathcal{D} = \{\mathbf{z}_i\}_{i=1}^N$ that contains N i.i.d. samples, where $\mathbf{z}_i = (\mathbf{x}_i, \mathbf{y}_i)$ with the input data $\mathbf{x}_i$ and the corresponding target label $\mathbf{y}_i$. We would like to learn a model with *weighs* $\boldsymbol{\omega}$ of the conditional distribution $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\omega})$, which could be softmax output of a classification neural network[1]. In the Bayesian perspective of learning, we would like to infer a *posterior* distribution over $\boldsymbol{\omega}$, i.e., $p(\boldsymbol{\omega}|\mathcal{D})$, after $\mathcal{D}$ is observed. Since computing the true posterior is intractable, we turn to optimize the parameters θ of some parametrized model $q_\theta(\boldsymbol{\omega})$, for example, a Gaussian model. The object of interest now becomes minimizing Kullback-Leibler (KL) divergence $\mathrm{KL}(q_\theta(\boldsymbol{\omega})\|p(\boldsymbol{\omega}|\mathcal{D}))$ in order to ensure $q_\theta(\boldsymbol{\omega})$ is a close approximation of $p(\boldsymbol{\omega}|\mathcal{D})$. A classical solution is to minimize a variational evidence lower bound (ELBO) $\mathcal{L}(\theta)$ as

$$\mathcal{L}(\theta) = -\mathrm{KL}(q_\theta(\boldsymbol{\omega})\|p(\boldsymbol{\omega})) + \mathcal{L}_{\mathcal{D}}(\theta), \tag{1}$$

[1]Note that the described Bayesian formula is also applicable to regression and other settings.

where $p(\boldsymbol{\omega})$ is a *prior* and the expected log-likelihood $\mathcal{L}_{\mathcal{D}}(\theta)$ is computed by

$$\mathcal{L}_{\mathcal{D}}(\theta) = \sum_{\mathbf{z}\in\mathcal{D}} \ell(\mathbf{z}, \theta) = \sum_{\mathbf{z}\in\mathcal{D}} \mathbb{E}_{q_\theta(\boldsymbol{\omega})}[\log p(\mathbf{y}|\mathbf{x}, \boldsymbol{\omega})]. \quad (2)$$

The uncertainty of BNN is hence captured by the variational distribution $q_\theta(\boldsymbol{\omega})$ which is often assumed to be a Gaussian $\mathcal{N}(\boldsymbol{\omega}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\theta = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ are mean and covariance parameters learned from data. This setting actually allows us to execute an *infinite ensemble* of models by sampling from $q_\theta(\boldsymbol{\omega})$, as the predictive distribution given a new sample $\mathbf{x}^*$ is characterized by

$$\begin{aligned} p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) &= \int p(\mathbf{y}^*|\mathbf{x}^*, \boldsymbol{\omega}) q_\theta(\boldsymbol{\omega}) d\boldsymbol{\omega} \\ &\simeq \frac{1}{T} \sum_t p(\mathbf{y}^*|\mathbf{x}^*, \boldsymbol{\omega}_t). \end{aligned} \quad (3)$$

The approximation term here indicates MC sampling, and T denotes the total number of sampled $\boldsymbol{\omega}_t$. With the inferred $p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D})$, we can quantify the uncertainty by calculating the *predictive variance* of output variable $\mathbf{y}^*$, denoted as $\mathrm{Var}(\mathbf{y}^*)$. We will provide details about its characterization in Section 3.3.

Nonetheless, this approach has major drawbacks: **(a).** The expectation term in Eq. (2), similar to practice in Eq. (3), requires MC sampling for approximation. This imposes T times more inference than DNN during optimization. **(b).** Apart from mean parameter $\boldsymbol{\mu}$, BNN pluses another covariance matrix $\boldsymbol{\Sigma}$ for optimizing $q_\theta(\boldsymbol{\omega})$. Suppose $\boldsymbol{\omega} \in \mathbb{R}^d$, then $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$ and the number of trainable parameters expands from d to $d + d^2$, which hinders if from large-scale NNs. In response to it, previous works usually simplify $q_\theta(\boldsymbol{\omega})$ to an isotropic or diagonal Gaussian, i.e., $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma} * I)$, while unavoidably yield coarse uncertainty quantification results due to the neglect of correlation between parameters. On account of this, we propose to employ an IJ-based method that does not require training models with additional memory and computational cost, and can quantify the uncertainty represented by full covariance.

3.2 Infinitesimal Jackknife for Pseudo Ensemble

In the frequentist notion of empirical risk minimization on $\mathcal{D}$, the optimal estimate of θ is defined by[2]

$$\theta^* = \arg\min_\theta \mathcal{L}_{\mathcal{D}}(\theta) = \arg\min_\theta \frac{1}{N} \sum_i \ell(\mathbf{z}_i, \theta). \quad (4)$$

In this scenario, instead of modifying DNN to BNN architecture, we propose to utilize *jackknife* (Efron and Stein 1981) for quantifying the uncertainty of the estimate. In particular, it drops one sample $\mathbf{z}_i$ out from the dataset $\mathcal{D}$ to build a leave-one-out subsample $\mathcal{D}_{\setminus i} \triangleq \mathcal{D} \setminus \{\mathbf{z}_i\}$, then re-fit the model to yield

$$\hat{\theta}_{\setminus i} = \arg\min_\theta \mathcal{L}_{\mathcal{D}_{\setminus i}}(\theta) \quad (5)$$

[2]In DNN, $q_\theta(\boldsymbol{\omega})$ reduces to a Dirac delta distribution, hence θ is now the mean variable $\boldsymbol{\mu}$ that has same size as $\boldsymbol{\omega}$.

as a jackknife estimate. Repeating this procedure for N times leads to a set of estimates $\{\hat{\theta}_{\setminus i}\}_{i=1}^N$, which can be used to compute the variance of θ^*. Specifically, we postulate $\hat{\theta}_{\setminus i}$ follows a multi-variate Gaussian distribution:

$$\begin{aligned} \hat{\theta}_{\setminus i} \sim \mathcal{N}(\boldsymbol{\mu}, \hat{\boldsymbol{\Sigma}}), \text{ where } \boldsymbol{\mu} &\triangleq \frac{1}{N} \sum_i \hat{\theta}_{\setminus i} = \theta^*, \\ \text{and } \hat{\boldsymbol{\Sigma}} &\triangleq \frac{1}{N} \sum_i (\hat{\theta}_{\setminus i} - \theta^*)(\hat{\theta}_{\setminus i} - \theta^*)^\top. \end{aligned} \quad (6)$$

It can be seen that similar to BNN, jackknife characterizes an ensemble of N models that represents the uncertainty about parameters and predictions.

Covariance approximation by IJ. By far, jackknife has not demonstrated its advantage since it is definitely intractable to retrain the model N times to compute the covariance matrix $\hat{\boldsymbol{\Sigma}}$ in practice. Here comes the infinitesimal jackknife, which succeeds in approximating $\hat{\theta}_{\setminus i}$ free of leave-one-out retraining. IJ is closely related to influence function (Huber 2004) which has been adopted in data quality measurement (Koh and Liang 2017; Wang et al. 2020). In particular, we up-weight $\mathbf{z}_i$ by a small ϵ to acquire the following optimization problem:

$$\hat{\theta}(\epsilon) = \arg\min_\theta \frac{1}{N} \sum_j \ell(\mathbf{z}_j, \theta) + \epsilon \ell(\mathbf{z}_i, \theta). \quad (7)$$

Note that removing $\mathbf{z}_i$ is now equivalent to $\epsilon = -\frac{1}{N}$. The main idea of influence function $\psi(\mathbf{z}_i)$ is to approximate $\hat{\theta}_{\setminus i}$ by minimizing the first-order Taylor series approximation around θ^* (Huber 2004), as

$$\psi(\mathbf{z}_i) \triangleq \left.\frac{d\hat{\theta}(\epsilon)}{d\epsilon}\right|_{\epsilon=0} = -H_{\theta^*}^{-1} \nabla_\theta \ell(\mathbf{z}_i, \theta^*), \quad (8)$$

where $H_{\theta^*} \triangleq \frac{1}{N}\sum_i \nabla_\theta^2 \ell(\mathbf{z}_i, \theta^*)$ denotes the Hessian matrix with θ^*. With the well-defined $\psi(\mathbf{z}_i)$, we can linearly approximate the jackknife estimate by $\hat{\theta}_{\setminus i} \simeq \theta^* - \frac{1}{N}\psi(\mathbf{z}_i)$, without retraining the model.

Therefore, plugging $\psi(\mathbf{z}_i)$ back into $\hat{\boldsymbol{\Sigma}}$ in Eq. (6) yields

$$\begin{aligned} \hat{\boldsymbol{\Sigma}} \simeq \frac{1}{N} \sum_i \frac{1}{N^2} \psi_i \psi_i^\top &= \frac{1}{N^2} H^{-1} \left[\frac{1}{N} \sum_i \nabla \ell_i \nabla \ell_i^\top\right] H^{-1} \\ &= \frac{1}{N^2} H^{-1} \mathcal{I}_{\theta^*} H^{-1}. \end{aligned} \quad (9)$$

Here ψ_i, $\nabla\ell_i$ and H are the abbreviation of $\psi(\mathbf{z}_i)$, $\nabla\ell(\mathbf{z}_i, \theta^*)$ and H_{θ^*} for avoiding clutter, respectively. $\mathcal{I}_{\theta^*}$ denotes the empirical Fisher information matrixt (FIM). At present, covariance $\hat{\boldsymbol{\Sigma}}$ in above Eq. (9) can be computed once the training process on full dataset $\mathcal{D}$ is done, instead of after N times leave-one-out retraining. Afterwards, we can sample $\boldsymbol{\omega} \sim \mathcal{N}(\theta^*, \hat{\boldsymbol{\Sigma}})$ during uncertainty inference.

Hessian approximation and efficient sampling. $\hat{\mathbf{\Sigma}}$ at hand, however, still raises concern of tractability due to the presence of the inverse Hessian matrix $H_{\theta^*}^{-1}$. Martens (2014) proposed a decomposition of it based on FIM as

$$H_{\theta^*} = \mathcal{I}_{\theta^*} + \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C}[\nabla_{\mathbf{y}^*}\ell(\mathbf{z}_i,\theta^*)]_j H_{[f]_j}, \tag{10}$$

where C is total number of classes, $\mathbf{y}^*$ is the output (or prediction) of the network given input $\mathbf{x}^*$, and $H_{[f]_j}$ is the Hessian of the j-th component of $\mathbf{y}^*$. In this situation, when almost all training samples are predicted correctly, as a well-trained network could achieve, we would have $\nabla_{\mathbf{y}^*}(\mathbf{z}_i,\theta^*) \simeq 0$ thus $H_{\theta^*} \simeq \mathcal{I}_{\theta^*}$. This result suggests that FIM is a stable positive semi-definite approximation of Hessian, hence $q_\theta(\boldsymbol{\omega})$ can be further cast to

$$q_\theta(\boldsymbol{\omega}) = \mathcal{N}(\theta^*, \hat{\mathbf{\Sigma}}), \text{ where } \hat{\mathbf{\Sigma}} = \frac{1}{N^2}\mathcal{I}_{\theta^*}^{-1}. \tag{11}$$

Furthermore, sampling from $q_\theta(\boldsymbol{\omega})$ can be done with precision matrix, i.e., $\hat{\mathbf{\Sigma}}^{-1} = N^2\mathcal{I}_{\theta^*}$, hence does not require taking inversion of $\mathcal{I}_{\theta^*}$. In detail, we first take an efficient Cholesky decomposition on the precision matrix $\mathbf{\Sigma}^{-1} = AA^\top$ where A is an upper triangular matrix. Then, we can sample $\boldsymbol{t} \sim \mathcal{N}(\mathbf{0}, I)$ and obtain the targeted $\boldsymbol{\omega}$ by

$$\boldsymbol{\omega} = \theta^* + \boldsymbol{t}' = \theta^* + A^{-1}\boldsymbol{t}. \tag{12}$$

Interestingly, we can circumvent from actually inverting A by employing a *back-substitution* trick that solves the special upper triangular system $A\boldsymbol{t}' = \boldsymbol{t}$ rather fast and stable.

Compared with the classical BNN that at most considers a diagonal covariance, our approach is capable of estimating the full covariance $\hat{\mathbf{\Sigma}}$ grounded on FIM, which is superior in terms of both efficiency and accuracy.

3.3 Quantifying Uncertainty of Variational Predictive Distribution

Currently, the weight distribution $q_\theta(\boldsymbol{\omega})$ has been built in Eq. (11), on which we are able to quantify the predictive uncertainty by MC sampling. Zhou et al. (2020) designed an uncertainty quantification regime that computed aleatoric and epistemic separately. By contrast, we advocate to characterize both uncertainties in a single model.

In detail, we start from the definition of variance of $q(\mathbf{y}^*)$, i.e., $\mathrm{Var}_{q(\mathbf{y}^*)}[\mathbf{y}^*]$. Denote $p(\mathbf{y}^*) \triangleq p(\mathbf{y}^*|\mathbf{x}^*,\boldsymbol{\omega})$, and the variational predictive distribution $q(\mathbf{y}^*) \triangleq p(\mathbf{y}^*|\mathbf{x}^*,\mathcal{D}) \simeq \frac{1}{T}\sum p(\mathbf{y}^*)$. The decomposition of $\mathrm{Var}_{q(\mathbf{y}^*)}[\mathbf{y}^*]$ can be conducted referring to the law of total variance (Kendall and Gal 2017) as

$$\begin{aligned}
&\mathrm{Var}_{q(\mathbf{y}^*)}[\mathbf{y}^*] = \mathbb{E}_{q(\mathbf{y}^*)}[\mathbf{y}^{*\otimes 2}] - \mathbb{E}_{q(\mathbf{y}^*)}[\mathbf{y}^*]^{\otimes 2} \\
&= \underbrace{\int \left\{\mathrm{diag}\{\mathbb{E}_{p(\mathbf{y}^*)}[\mathbf{y}^*]\} - \mathbb{E}_{p(\mathbf{y}^*)}[\mathbf{y}^*]^{\otimes 2}\right\} q_\theta(\boldsymbol{\omega}) d\boldsymbol{\omega}}_{\text{aleatoric}} \\
&+ \underbrace{\int \left\{\mathbb{E}_{p(\mathbf{y}^*)}[\mathbf{y}^*] - \mathbb{E}_{q(\mathbf{y}^*)}[\mathbf{y}^*]\right\}^{\otimes 2} q_\theta(\boldsymbol{\omega}) d\boldsymbol{\omega}}_{\text{epistemic}},
\end{aligned} \tag{13}$$

where $v^{\otimes 2} = vv^\top$, $\mathrm{diag}(v)$ represents a diagonal matrix whose element vector is v. Based on this decomposition, Kwon et al. (2020) further proposed an elegant solution as[3]

$$\underbrace{\frac{1}{T}\sum_{t=1}^{T}\mathrm{diag}(\hat{\mathbf{y}}_t) - \hat{\mathbf{y}}_t^{\otimes 2}}_{\hat{\mathbf{\Sigma}}_{\text{alea}}} + \underbrace{\frac{1}{T}\sum_{t=1}^{T}(\hat{\mathbf{y}}_t - \bar{\mathbf{y}})^{\otimes 2}}_{\hat{\mathbf{\Sigma}}_{\text{epis}}}, \tag{14}$$

where $\hat{\mathbf{y}}_t \triangleq p(\mathbf{y}^*|\mathbf{x}^*,\boldsymbol{\omega}_t)$ and $\bar{\mathbf{y}} \triangleq \frac{1}{T}\sum \hat{\mathbf{y}}_t$. This method directly obtains the variability from predictions during MC sampling, and does not involve sampling steps on the additional variance terms as done in Kendall and Gal (2017).

One more merit drawn from Eq. (14) is the clear clarification of aleatoric and epistemic quantities. It allows room for evaluating sample difficulty in two orthogonal aspects: while aleatoric uncertainty is a measure of the data variation, epistemic measures the model variation. We will present how to make use of it in the next section.

4 Framework

We have addressed uncertainty quantification by IJ, in this section, we turn to present the framework of uncertainty-guided CL (UCL) based on it. We introduce the implemented framework in two most important components in CL: (1) sample difficulty scoring and (2) pacing the sample presentation during training (or curriculum arrangement). Then, we conclude the overall procedure in an algorithm.

4.1 Sample Difficulty Scoring

Using confidence of a specific classifier, e.g., the margin of a support vector machine (SVM) classifier, has been agreed as a reasonable indicator for sample difficulty (Weinshall, Cohen, and Amir 2018). However, training SVM on a large dataset can be time-consuming, even if on the extracted activations from the penultimate layer of a pretrained network, namely knowledge transfer suggested in Hacohen and Weinshall (2019). By contrast, our method can quantify the model confidence by IJ without extra classifier learning. We present the proposed scoring function on the following two aspects.

Uncertainty knowledge. Inspired by the knowledge transfer strategy, we investigate two approaches for uncertainty quantification: (1) self-tutoring and (2) uncertainty transfer. In self-tutoring, we first train the network from scratch without curriculum, then quantify the uncertainty of the same network by IJ and infer the predictive uncertainty on training data. On the other, we involve another pretrained network and finetune it on the dataset at hand, then use it to infer data uncertainty. Notably, these two methods only differ in the model utilized for inference in sample scoring.

Difficulty metric. Based on the two factorized variance $\hat{\mathbf{\Sigma}}_{\text{alea}}$ and $\hat{\mathbf{\Sigma}}_{\text{epis}}$ in Eq. (14), it is reasonable to design three scoring metrics, aleatoric-based Ω_{alea}, epistemic-based Ω_{epis}, and the total variance $\Omega_{\text{total}} = \Omega_{\text{alea}} + \Omega_{\text{epis}}$. Intuitively, the higher the three metrics are, the harder a sample for the model to make decisions. To obtain the values Ω_* from

[3] "alea" and "epis" are shorthands of aleatoric and epistemic.

Algorithm 1 Uncertainty-guided CL via IJ.

Require: Training set $\mathcal{D}$; Teacher model f_t and student model f_s; Number of baby steps S.

1: Train f_t on $\mathcal{D}$, then compute the FIM $\mathcal{I}_{\theta^*}$ to adapt f_t to a Bayesian neural network (Section 3.2).
2: Compute predictive uncertainty for each $\mathbf{z}$ in $\mathcal{D}$ using the Bayesian f_t by MC sampling (Section 3.3).
3: Compute sample scores Ω (Section 4.1), then split $\mathcal{D}$ into S subsets $\{\mathcal{D}_1, \ldots, \mathcal{D}_S\}$ according to Ω.
4: Initialize cumulative dataset $\mathcal{D}^* = \emptyset$.
5: **for** epoch $s = 1 \rightarrow S$ **do**
6: Aggregate $\mathcal{D}_s$ into $\mathcal{D}^*$, i.e., $\mathcal{D}^* \leftarrow \mathcal{D}^* \cup \mathcal{D}_s$.
7: Train student model f_s on $\mathcal{D}^*$.
8: **end for**

$C \times C$ matrix $\hat{\mathbf{\Sigma}}_*$ for ranking samples, we simply pick the (k, k)-th element from $\hat{\mathbf{\Sigma}}_*$ which corresponds to the groundtruth label of the sample, i.e., $\mathbf{y}$ is a one-hot vector where only its k-th element equals one. This approach empirically reaches satisfying results. Moreover, inspired by Blundell et al. (2015) that takes into account both mean and standard deviation of weight parameters, we propose a SNR metric that combines the magnitude of prediction corresponding to the groudtruth class $\hat{\mathbf{y}}^k$ and uncertainty Ω_{total} together, as

$$\Omega_{\text{snr}} \triangleq \frac{\hat{\mathbf{y}}^k}{\Omega_{\text{total}}}. \tag{15}$$

SNR is a well-known measure in signal processing to distinguish between useful information from noise contained in signal. In this context, Ω_{snr} indicates the model capacity of identifying groundtruth class from data and model uncertainty; the higher the Ω_{snr}, the easier a sample for the model making prediction.

4.2 Curriculum Arrangement

Hacohen and Weinshall (2019) investigated three typical pacing functions, including exponential pacing, varied exponential pacing and single-step pacing, and found all three have comparable performance. Considering this, we adopt a *baby step* (Cirik, Hovy, and Morency 2016) regime to arrange training data based on their scores. We split the training set $\mathcal{D}$ into S buckets $\{\mathcal{D}_1, \ldots, \mathcal{D}_S\}$ with respect to the calculated sample scores Ω. The training starts from the easiest sample set $\mathcal{D}_1$, then aggregates $\mathcal{D}_2$ and $\mathcal{D}_1$ in the next step, and so on until the full set is included. Notably the single-step pacing is a special case of baby step when $S = 2$.

The main procedure of our algorithm is presented in Algorithm 1. Note that the teacher is the same model as the student in self-tutoring, while in uncertainty transfer, the teacher model is another pretrained network, e.g., ResNet (He et al. 2016).

5 Experiments

We evaluate our UCL method and a series of CL baselines on many image and text datasets, in order to demonstrate the merit of the proposed method.

Table 1: Summary statistics of datasets.

Dataset	# Training	# Test	# Classes
CIFAR-10	50,000	10,000	10
CIFAR-100	50,000	10,000	20/100
STL-10	50,000	80,000	10
SVHN	73,257	26,032	10
Ohsumed	5,180	2,220	23
R52	6,370	2,730	52
MR	7,464	3,198	2
20NG	13,193	5,653	20

5.1 Datasets

Our UCL method is capable of handling both image and text data, such that we conduct experiments on four widely used image classification datasets: **CIFAR-10** & **CIFAR-100** (Krizhevsky, Hinton, and others 2009), **STL-10** (Coates, Ng, and Lee 2011), **SVHN** (Netzer et al. 2011); and four text classification benchmark datasets: **20NG**[4], **MR**[5], **R52**[6], **Ohsumed**[7]. For the image datasets, we follow the training/test split of the original release. For the text datasets, we follow the preprocessing steps in Yao, Mao, and Luo (2019) but differ in dataset split. We randomly draw 70% data from the raw corpus as training set, and the rest 30% as test set. During training, we take 10% data from the training set as validation set for searching hyperparameters. A summary of the dataset statistics is shown in Table 1.

5.2 Baselines

We include a variety of state-of-the-art baselines in comparison against the proposed UCL method:

- **No-CL**. This is the student model trained without curriculum learning. It is the controlled group that evaluates absolution gain led by CL methods.
- **SPL, SPCL and SPL-IR** (Kumar, Packer, and Koller 2010; Jiang et al. 2015; Fan et al. 2017). Self-paced learning is a pioneering work on predefined CL, then come a series follow-ups like self-paced curriculum learning (SPCL) and self-paced learning with implicit regularization (SPL-IR), which try to derive effective and robust data-driven curriculums.
- **CurriculumNet** (Guo et al. 2018). CurriculumNet leverages unsupervised clustering algorithm to evaluate data complexity by its distribution density. Before clustering, the raw images are encoded by a strong teacher model, e.g., ResNet50, as embedding vectors. Note that the teacher model should be fine-tuned on the task at hand.
- **MentorNet** (Jiang et al. 2018). It adopts a teacher model that specifically focuses on learning to weight samples.

[4] http://qwone.com/~jason/20Newsgroups/

[5] http://www.cs.cornell.edu/people/pabo/movie-review-data/

[6] https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection

[7] http://disi.unitn.it/moschitti/corpora.htm

Table 2: Test accuracy on image and text classification tasks of the compared baselines and our UCL methods, where the best ones are in bold.

	Image				Text			
Method	**CIFAR-10**	**CIFAR-100**	**STL-10**	**SVHN**	**Ohsumed**	**R52**	**MR**	**20NG**
No-CL	0.6137	0.3678	0.4935	0.5789	0.3387	0.7304	0.6891	0.5821
SPL	0.6177	0.3741	0.5021	0.5794	0.3310	0.7300	0.6907	0.5839
SPCL	0.6193	0.3731	0.5076	0.5835	0.3229	0.7296	0.6916	0.5988
SPL-IR	0.6079	0.3569	0.5062	0.5853	0.3108	0.7135	0.6094	0.5444
CL-TL	0.6283	0.3980	0.5242	0.7637	0.3090	0.7315	0.6548	0.3384
MentorNet	0.6248	0.3700	0.4977	0.5853	0.3374	0.7106	0.6994	0.5052
CurriculumNet	0.6014	0.3652	0.5220	0.5053	0.3130	0.6656	0.6645	0.4466
Data-Param	0.6182	0.3459	0.5158	0.6311	0.3423	0.7479	0.6929	0.5926
UCL-TL	0.6265	**0.3986**	**0.5263**	0.7771	0.3261	0.7099	0.6695	0.4528
UCL-alea	0.6300	0.3843	0.5073	0.7984	0.3572	0.7579	0.7142	0.6274
UCL-epis	0.6279	0.3880	0.5167	**0.8065**	0.3558	0.7597	0.7104	**0.6292**
UCL-total	0.6251	0.3980	0.5164	0.7816	0.3577	0.7582	**0.7170**	0.6262
UCL-snr	**0.6306**	0.3956	0.5205	0.7869	**0.3586**	**0.7645**	0.7133	0.6158

We need to train a student model first, meanwhile record the states of it during training process, e.g., loss quantile, epochs, etc., then use these records to supervise learning of the teacher model. Finally, we train the student collaboratively with the teacher model.

- **CL-TL** (Weinshall, Cohen, and Amir 2018; Hacohen and Weinshall 2019). Curriculum learning by transfer learning advocates to score sample difficulty with knowledge transfer. Like curriculumNet, it encodes raw images into embeddings with fine-tuned teacher model, while it trains a SVM classifier on these embeddings and scores samples with the confidence of SVM.
- **DataParam** (Saxena, Tuzel, and DeCoste 2019). This method is basically a variant of SPL by equipping each sample and class with a learnable parameter, namely data parameters.
- **UCL-TL**. In Section 4.1 we mention that UCL can utilize a powerful teacher model to infer data uncertainty, namely UCL-TL. It is for exploring the gain from uncertainty knowledge transfer in UCL.
- **UCL-alea, -epis, -total and -snr**. They are our UCL method associated with different uncertainty metrics: Ω_{alea}, Ω_{epis}, Ω_{total} and Ω_{snr}, referring to Section 4.1.

5.3 Experimental Protocol

Network architecture. We use two different protocols of student and teacher models for image and text datasets, respectively. For image datasets, the student model is a convolutional neural network (CNN) with one convolution layer and two dense layers; the teacher model is ResNet50 with its last dense layer adapted to specific number of classes in each task. For text datasets, the student model is TextCNN (Kim 2014) with three convolution layers and one dense layer; the teacher model is a frozen BERT-base (Devlin et al. 2019) model attached with two trainable dense layers. Notably, the student and teacher model with same hyperparameters are used in all methods to ensure a fair comparison.

Table 3: Test accuracy after the initial stage and the final result on SVHN. C-Net is the shorthand of CurriculumNet.

Method	**No-CL**	**CL-TL**	**C-Net**	**UCL-TL**	**UCL**
Initial	0.5601	0.6149	0.4854	0.6898	**0.6995**
Final	0.5783	0.7637	0.5053	0.7771	**0.8105**

Pacing function. It has been specified that pacing function plays less important role than scoring function in CL. As this paper mainly presses on sample scoring, we take baby step with two stages (single-step pacing) for CurriculumNet, CL-TL, UCL-TL and UCL: 20% data are involved at the first stage, then all data are used at the second stage.

5.4 Test Results

Overall results are demonstrated in Table 2. Two main findings can be drawn as the following:

- No method **except for** self-tutoring UCL consistently outperforms No-CL on all datasets. Apart from CIFAR-100 and STL-10, UCL reaches the best results on all the rest datasets, especially demonstrates huge margin on SVHN. Overall, UCL-total is better than only using Ω_{alea} or Ω_{epis}. And UCL-snr further improves over them. In particular, UCL-TL does not gain significant improvement over UCL accounting for the additional consumption in transfer learning.
- The first training stage almost determines the final results in CL, which is aligned with Hacohen and Weinshall (2019) that empirically revealed most of the power of CL lies at the beginning of training. As shown in Table 3, model is trained with 20% data in the initial stage and then trained on the whole dataset. It could be observed a better starting point usually enables better final result.

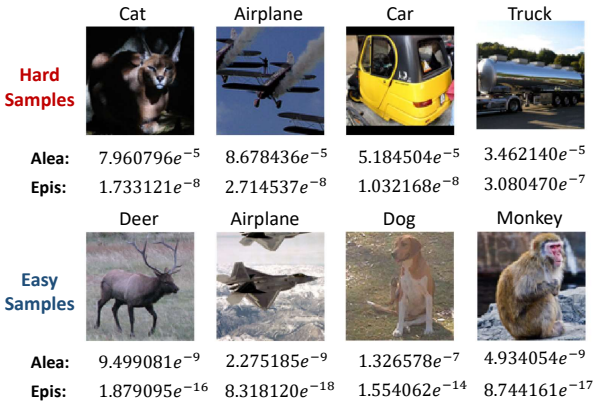

Figure 1: Finding influential instances within a bag via subsampling based on the calculated probability π. Note that here *negative* ϕ means a beneficial sample.

- Although transfer learning based methods often obtain good results on image data, they actually *impair* learning on text data empirically. The cause might be that the teacher BERT+MLP model has a different view of sample difficulty from the student TextCNN model. On the other hand, self-tutoring UCL is consistent in rationale of uncertainty, since it is measured by the student model itself.

We value predictive uncertainty of models because it touch the and sheds light on interpretability in CL, as well as its further prospects, e.g., outlier detection, noisy label detection and open world recognition.

References

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 41–48.

Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; and Wierstra, D. 2015. Weight uncertainty in neural networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, 1613–1622.

Cirik, V.; Hovy, E.; and Morency, L.-P. 2016. Visualizing and understanding curriculum learning for long short-term memory networks. *arXiv preprint arXiv:1611.06204*.

Coates, A.; Ng, A.; and Lee, H. 2011. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth International Conference on Artificial Intelligence and Statistics*, 215–223.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Dogan, U.; Deshmukh, A. A.; Machura, M.; and Igel, C. 2019. Label-similarity curriculum learning. *arXiv preprint arXiv:1911.06902*.

Efron, B., and Stein, C. 1981. The jackknife estimate of variance. *The Annals of Statistics* 586–596.

Fan, Y.; He, R.; Liang, J.; and Hu, B. 2017. Self-paced learning: an implicit regularization perspective. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Gal, Y., and Ghahramani, Z. 2015. Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*.

Gal, Y., and Ghahramani, Z. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, 1050–1059.

Gal, Y.; Islam, R.; and Ghahramani, Z. 2017. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, 1183–1192.

Graves, A. 2011. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems*, 2348–2356.

Guo, S.; Huang, W.; Zhang, H.; Zhuang, C.; Dong, D.; Scott, M. R.; and Huang, D. 2018. Curriculumnet: Weakly supervised learning from large-scale web images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 135–150.

Hacohen, G., and Weinshall, D. 2019. On the power of curriculum learning in training deep networks. In *International Conference on Machine Learning*, 2535–2544.

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

Hernández-Lobato, J. M., and Adams, R. 2015. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International Conference on Machine Learning*, 1861–1869.

Hinton, G. E., and Van Camp, D. 1993. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth Annual Conference on Computational Learning Theory*, 5–13.

Huber, P. J. 2004. *Robust statistics*, volume 523. John Wiley & Sons.

Jaeckel, L. A. 1972. *The infinitesimal jackknife*. Bell Telephone Laboratories.

Jiang, L.; Meng, D.; Zhao, Q.; Shan, S.; and Hauptmann, A. G. 2015. Self-paced curriculum learning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Jiang, L.; Zhou, Z.; Leung, T.; Li, L.-J.; and Fei-Fei, L. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, 2304–2313.

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*, 5574–5584.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–1751.

Koh, P. W., and Liang, P. 2017. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, 1885–1894.

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

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

Kumar, M. P.; Packer, B.; and Koller, D. 2010. Self-paced learning for latent variable models. In *Advances in neural information processing systems*, 1189–1197.

Kwon, Y.; Won, J.-H.; Kim, B. J.; and Paik, M. C. 2020. Uncertainty quantification using bayesian neural networks in classification: Application to biomedical image segmentation. *Computational Statistics & Data Analysis* 142:106816.

MacKay, D. J. 1992. A practical bayesian framework for backpropagation networks. *Neural computation* 4(3):448–472.

Martens, J. 2014. New insights and perspectives on the natural gradient method. *arXiv preprint arXiv:1412.1193*.

Neal, R. M. 2012. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning.

Nguyen, C. V.; Li, Y.; Bui, T. D.; and Turner, R. E. 2018. Variational continual learning. In *International Conference on Learning Representations*.

Peterson, G. B. 2004. A day of great illumination: Bf skinner's discovery of shaping. *Journal of the experimental analysis of behavior* 82(3):317–328.

Saxena, S.; Tuzel, O.; and DeCoste, D. 2019. Data parameters: A new family of parameters for learning a differentiable curriculum. In *Advances in Neural Information Processing Systems*, 11095–11105.

Skinner, B. F. 1958. Reinforcement today. *American Psychologist* 13(3):94.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Wang, Z.; Zhu, H.; Dong, Z.; He, X.; and Huang, S.-L. 2020. Less is better: Unweighted data subsampling via influence function. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 6340–6347.

Wang, Y.; Kucukelbir, A.; and Blei, D. M. 2017. Robust probabilistic modeling with bayesian data reweighting. In *International Conference on Machine Learning*, 3646–3655.

Weinshall, D.; Cohen, G.; and Amir, D. 2018. Curriculum learning by transfer learning: Theory and experiments with deep networks. In *International Conference on Machine Learning*, 5238–5246.

Yao, L.; Mao, C.; and Luo, Y. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 7370–7377.

Zhou, Y.; Yang, B.; Wong, D. F.; Wan, Y.; and Chao, L. S. 2020. Uncertainty-aware curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6934–6944.