Variational Learning of Bayesian Neural Networks via Bayesian Dark Knowledge

Gehui Shen, **Xi Chen** and **Zhihong Deng***
Key Laboratory of Machine Perception (Ministry of Education),
School of Electronics Engineering and Computer Science, Peking University
{jueliangguke, mrcx, zhdeng}@pku.edu.cn

Abstract

Bayesian neural networks (BNNs) have received more and more attention because they are capable of modeling epistemic uncertainty which is hard for conventional neural networks. Markov chain Monte Carlo (MCMC) methods and variational inference (VI) are two mainstream methods for Bayesian deep learning. The former is effective but its storage cost is prohibitive since it has to save many samples of neural network parameters. The latter method is more time and space efficient, however the approximate variational posterior limits its performance. In this paper, we aim to combine the advantages of above two methods by distilling MCMC samples into an approximate variational posterior. On the basis of an existing distillation technique we first propose variational Bayesian dark knowledge method. Moreover, we propose Bayesian dark prior knowledge, a novel distillation method which considers MCMC posterior as the prior of a variational BNN. Two proposed methods both not only can reduce the space overhead of the teacher model so that are scalable, but also maintain a distilled posterior distribution capable of modeling epistemic uncertainty. Experimental results manifest our methods outperform existing distillation method in terms of predictive accuracy and uncertainty modeling.

1 Introduction

In the past few years, deep learning has achieved great success in many fields. However, deep neural networks (DNNs) also have some drawbacks. DNNs typically need large labeled datasets to prevent overfitting so are incapable in scarce data scenarios. Moreover, a more serious problem is DNNs typically give overconfident predictive distribution even for the *out of distribution* (OOD) data they have not seen before [Gal, 2016] and are poorly calibrated [Guo *et al.*, 2017]. For example, a high softmax probability does not mean high confidence in classification. In some scenarios related to human safety, such as autonomous vehicles and automated disease detection systems, knowing what the model does not know is crucial to preventing undesirable behaviour. To meet this requirement, we can resort to *epistemic* uncertainty, which accounts for uncertainty in the model parameters and can be explained away given enough data [Kendall and Gal, 2017]. In addition, there are many machine learning applications which rely on epistemic uncertainty to make decision, such as active learning and deep reinforcement learning [Gal, 2016; Gal *et al.*, 2017; Depeweg *et al.*, 2017].

Bayesian inference is a principled approach to tackle the aforementioned overfitting and overconfidence problem in DNNs. In Bayesian deep learning (BDL) framework, we place a prior over DNNs' weights and infer a posterior distribution over the weights given some data. This type of model is called as Bayesian neural networks (BNNs) [MacKay, 1992; Hinton and van Camp, 1993; Neal, 1995]. During test, we obtain prediction distribution by marginalizing over the posterior distribution. The uncertainty of parameters, i.e. *epistemic* uncertainty results in reasonable uncertainty about prediction.

Recently, with the popularity of deep learning, BNNs have witnessed a revival. Markov Chain Monte Carlo (MCMC) [Welling and Teh, 2011; Chen *et al.*, 2014; Ding *et al.*, 2014] and variational inference (VI) [Graves, 2011; Blundell *et al.*, 2015; Louizos and Welling, 2016; Louizos and Welling, 2017] are two most general methods for modern BNNs inference. In the limit of time, MCMC methods can generate samples from the true posterior asymptotically and thus make more accurate prediction. The main drawback of MCMC methods is the prohibitive storage cost. Since the posterior distribution is represented by samples of DNN parameters, we have to save thousands of or even more samples. For modern DNNs which generally have millions or ten millions of parameters this approach may be not feasible. In addition, during test, the time cost is also high because we should evaluate the model one pass for each sample. In contrast, VI methods are more time and space efficient but the gap between the approximate posterior and the true posterior degenerates the model performance.

Hinton *et al.* [2014] have proposed "distillation" training framework which aims to transfer the knowledge from a cumbersome model into a small model which is easier to deploy. This idea has already been adapted to compress MCMC-based BNNs. Bayesian Dark Knowledge (BDK) [Balan *et al.*, 2015] method trains a non-Bayesian student DNN to distill a teacher BNN which is trained with Stochastic Gradient Langevin Dy-

*Corresponding author

namics (SGLD) [Welling and Teh, 2011], a typical MCMC method for BDL, as the original distillation paper does. However, the student network is not a BNN so that it loses the ability of teacher to model epistemic uncertainty as there is no posterior distribution representing the uncertainty of weights. It is a non-trivial drawback as epistemic uncertainty estimation is an important feature of BNNs as above discussed.

To overcome this issue, Wang *et al.* [2018] have proposed another method, called Adversarial Posterior Distillation (APD) with Generative Adversarial Networks (GANs) [Goodfellow *et al.*, 2014]. They exploit a GAN to directly distill the SGLD samples into a generator. After training, the generator is expected to produce samples from SGLD posterior approximately. It should be noted that APD method has a fatal disadvantage that as the output dimension of generator and the input dimension of discriminator are just the number of the weights in BNNs, with the hidden size set to 100 [Wang *et al.*, 2018], the number of parameters of the whole model is two hundred times that of the BNN. The prohibitive storage cost problem of MCMC methods still exists thus limits its scalability. Hence we think it is the main drawback of APD as it partially loses the essential feature of distillation.

In this paper we propose a new distillation framework for BDL to tackle the drawbacks of BDK and APD simultaneously. Overall speaking, in our distillation framework, the teacher network is a BNN trained by a MCMC sampler while the student network is also a BNN with the same size as teacher but is optimized with variational learning. The motivation of such a design is three-fold: Firstly, as MCMC methods always lead to accurate posterior approximation, we intend to transfer the knowledge hidden in the MCMC samples into small models to get rid of its prohibitive storage cost. Secondly, to enable the student network to model epistemic uncertainty, BNNs with an posterior distribution are more appealing than DNNs with point estimate. Thirdly, training a BNN with variational inference can only need storage overhead that is several times the number of network weights, which guarantees the scalability.

Our contributions are as follows: Firstly, to save the storage overhead of MCMC for BDL, we make the first attempt to combine VI and MCMC techniques for BDL by distilling the knowledge in MCMC samples into a variational BNN, from which we can still draw samples for epistemic uncertainty prediction. Secondly, we propose variational Bayesian dark knowledge method on the basis of BDK, as well as a novel distillation method, named Bayesian dark prior knowledge, which treats the knowledge in the teacher as prior to constrain the variational objective of the student. Finally, experimental results show that proposed methods are scalable and perform better on both predictive accuracy and uncertainty modeling than existing distillation methods.

2 Preliminaries: Bayesian Neural Networks

In BNNs, the network weights $\mathbf{w}$ are considered as random variables. Given the training data $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$, we need to calculate the posterior distribution of weights with Bayes rule:

$$p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{\int p(\mathcal{D}|\mathbf{w})p(\mathbf{w})d\mathbf{w}} \quad (1)$$

where the first term in the numerator $p(\mathcal{D}|\mathbf{w}) = \prod_{n=1}^N p(\mathbf{y}_n|\mathbf{x}_n, \mathbf{w})$ is the likelihood and the second term represents the prior on the weights. To test an unseen data $\hat{\mathbf{x}}$, the predictive distribution of the label $\hat{\mathbf{y}}$ is given by $p(\hat{\mathbf{y}}|\hat{\mathbf{x}}) = \mathbb{E}_{p(\mathbf{w}|\mathcal{D})}[p(\hat{\mathbf{y}}|\hat{\mathbf{x}}, \mathbf{w})]$. The uncertainty of weights due to the posterior distribution allows us to model uncertainty about data samples. However, the Eq (1) is intractable because of the high-dimensional integral in the denominator, so we have to resort to approximate methods.

2.1 VI for Bayesian Neural Networks

Variational Bayes is a general approximate method to inference and learning in Bayesian models. For BNNs, VI finds a variational approximation to the Bayesian posterior distribution on the weights [Hinton and van Camp, 1993]. Given a parametric variational posterior $q(\mathbf{w}|\theta)$ with parameters θ, VI minimizes the Kullback-Leibler (KL) divergence between $q(\mathbf{w}|\theta)$ and $p(\mathbf{w}|\mathcal{D})$:

$$\begin{aligned}\theta^\star &= \arg\min_\theta \mathrm{KL}[q(\mathbf{w}|\theta)\|p(\mathbf{w}|\mathcal{D})] \\ &= \arg\min_\theta(-\mathbb{E}_{q(\mathbf{w}|\theta)}[\log p(\mathcal{D}|\mathbf{w})] + \mathrm{KL}[q(\mathbf{w}|\theta)\|p(\mathbf{w})])\end{aligned} \quad (2)$$

The final cost function is known as negative evidence lower bound (ELBO). *Bayes by Backprop* (BBB) [Blundell *et al.*, 2015] is a simple but efficient VI method for BDL. By using reparametrization trick [Kingma and Welling, 2014] BBB can make unbiased Monte Carlo gradient estimator for the first term in the negative ELBO:

$$\mathbb{E}_{q(\mathbf{w}|\theta)}[\log p(\mathcal{D}|\mathbf{w})] = \mathbb{E}_{p(\epsilon)}[\log p(\mathcal{D}|f(\theta, \epsilon))]. \quad (3)$$

BBB further supposes that $q(\mathbf{w}|\theta)$ is a diagonal Gaussian distribution and each weight w_i is parametrized by a mean μ_i and a standard deviation σ_i which can be reparametrized by:

$$w_i = f(\theta, \epsilon_i) = \mu_i + \sigma_i \odot \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, 1) \quad (4)$$

with the standard Gaussian prior, the second KL term in the negative ELBO can be computed analytically:

$$\mathrm{KL}[q(\mathbf{w}|\theta)\|p(\mathbf{w})] = \frac{1}{2}\sum_i(\sigma_i^2 + \mu_i^2 - \log\sigma_i^2 - 1) \quad (5)$$

2.2 Stochastic Gradient Langevin Dynamics

SGLD [Welling and Teh, 2011] is a stochastic gradient MCMC method for scalable Bayesian learning. By injecting a noise into stochastic gradient descent (SGD) update, the algorithm can simulate samples from the posterior using Langevin dynamics in an MCMC manner. With theoretical guarantee, the SGLD update can be conducted in a mini-batch manner:

$$\begin{aligned}\Delta\mathbf{w}_t = \frac{\eta_t}{2}\left(\nabla\log p\left(\mathbf{w}_t\right) + \frac{N}{M}\sum_{i=1}^M \nabla\log p\left(\mathbf{y}_{t_i}|\mathbf{x}_{t_i}, \mathbf{w}_t\right)\right) \\ + \mathbf{z}_t \quad \mathbf{z}_t \sim \mathcal{N}\left(0, \eta_t I\right)\end{aligned} \quad (6)$$

where M is the mini-batch size and N is the training data size. t indexes the mini-batch iterations with gradient step size η_t. If we use the standard Gaussian prior, the SGLD update is just the SGD update for a L2 regularized neural network with added

Gaussian noise $\mathbf{z}_t$. SGLD represents the posterior distribution by Monte Carlo samples: $q_{SGLD}(\mathbf{w}) = \frac{1}{S}\sum_{s=1}^{S}\delta(\mathbf{w}-\mathbf{w}_s)$ where S is the number of samples, instead of a parametric model. At test time, $p(\hat{\mathbf{y}}|\hat{\mathbf{x}}) = \frac{1}{S}\sum_{s=1}^{S} p(\hat{\mathbf{y}}|\hat{\mathbf{x}}, \mathbf{w}_s)$. Such an approximation results in S times storage cost. It is easy to become infeasible for DNNs with millions of parameters.

3 Methods

In our methods, we regard MCMC sampler as a teacher model $\mathcal{T}$ and distill the knowledge in it into a student variatioanl BNN $\mathcal{S}$. Compared to the DNN student of BDK [Balan *et al.*, 2015], our BNN student possesses a posterior distribution and is therefore suitable for model epistemic uncertainty. We employ SGLD to draw MCMC samples for distillation following previous works [Balan *et al.*, 2015; Wang *et al.*, 2018] and choose BBB [Blundell *et al.*, 2015] to train student $\mathcal{S}$ for the sake of the low storage overhead and easy implementation. Our framework is also compatible with extensions to SGLD and more sophisticated VI methods. We first adapt the negative ELBO in Eq (2) to mini-batch optimization for student $\mathcal{S}$ with reparameterization trick:

$$\begin{aligned}\mathcal{L}(\theta|\mathcal{M}) = &-\frac{1}{M}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{M}}\mathbb{E}_{p(\epsilon)}[\log p(\mathbf{y}|\mathbf{x}, f(\theta,\epsilon))] \\ &+\frac{1}{N}\mathrm{KL}[q(\mathbf{w}|\theta)\|p(\mathbf{w})]\end{aligned} \tag{7}$$

where M is the size of mini-batch $\mathcal{M}$ and N is the training data size. Two proposed methods introduce the knowledge from teacher through two terms in Eq (7) respectively.

3.1 Variational Bayesian Dark Knowledge

BDK method trains the student $\mathcal{S}$ to approximate the predictive distribution of the teacher $\mathcal{T}$ and thus the objective is $\mathrm{KL}[\mathcal{T}(\mathbf{y}|\mathbf{x})\|\mathcal{S}(\mathbf{y}|\mathbf{x},\theta)] = -\mathbb{E}_{\mathcal{T}(\mathbf{y}|\mathbf{x})}\log\mathcal{S}(\mathbf{y}|\mathbf{x},\theta) + \text{const}$. In practice, BDK conducts online learning and the predictive distribution is approximately computed by a single posterior sample of SGLD, i.e. $\mathcal{T}(\mathbf{y}|\mathbf{x}) \approx p(\mathbf{y}|\mathbf{x},\mathbf{w}'), \mathbf{w}' \sim q_{SGLD}(\mathbf{w})$. Inspired by this objective, we replace the likelihood term with an expected KL term:

$$\begin{aligned}&\mathbb{E}_{p(\epsilon)}\big[\mathrm{KL}[\mathcal{T}(\mathbf{y}|\mathbf{x})\|p(\mathbf{y}|\mathbf{x}, f(\theta,\epsilon))]\big] \\ &= -\mathbb{E}_{p(\epsilon)}[\mathbb{E}_{p(\mathbf{y}|\mathbf{x},\mathbf{w}_t)}\log p(\mathbf{y}|\mathbf{x}, f(\theta,\epsilon))] + \text{const}\end{aligned} \tag{8}$$

For classification, the inner expectation can be seen as the cross-entropy loss with the soft label $\hat{\mathbf{y}} \sim p(\mathbf{y}|\mathbf{x},\mathbf{w}')$, the softmax distribution predicted by the teacher. For simplicity we denote $\mathbb{E}_{p(\mathbf{y}|\mathbf{x},\mathbf{w}')}\log p(\mathbf{y}|\mathbf{x}, f(\theta,\epsilon))$ by $\log p(\hat{\mathbf{y}}|\mathbf{x}, f(\theta,\epsilon))$. Therefore, we obtain an objective conditioned on an SGLD sample for training student:

$$\begin{aligned}\mathcal{L}'(\theta|\mathcal{M}',\mathbf{w}') = &-\frac{1}{M}\sum_{\mathbf{x}\in\mathcal{M}'}\mathbb{E}_{p(\epsilon)}[\log p(\hat{\mathbf{y}}|\mathbf{x}, f(\theta,\epsilon))] \\ &\frac{1}{N}\mathrm{KL}[q(\mathbf{w}|\theta)\|p(\mathbf{w})]\end{aligned} \tag{9}$$

Compared to standard ELBO in Eq (7), the new variational objective only replaces gold one-hot label with soft label predicted by the teacher. Therefore we obtain a simple and principled method for training student BNNs by extending original BDK method. In order to avoid confusion with BDK, we name this method Variational BDK (V-BDK).

During training, at iteration t we sample a mini-batch $\mathcal{M} = \{(\mathbf{x}_{t_i}, \mathbf{y}_{t_i})\}_{i=1}^{M}$ to obtain a single SGLD sample $\mathbf{w}_t$. To make student generalize better, we generate new data $\{\mathbf{x}'_{t_i}\}$ "near" the training data $\{\mathbf{x}_{t_i}\}$ by adding slight noise [Hinton *et al.*, 2014]. The labels of new data are softmax distributions predicted by the teacher. We optimize student with the noisy mini-batch and the objective is the revised negative ELBO $\mathcal{L}'(\theta|\mathcal{M}',\mathbf{w}_t)$ in Eq (9). As V-BDK is an online learning algorithm like BDK, we only need to save one SGLD sample which is space efficient.

3.2 Bayesian Dark Prior Knowledge

Since the teacher and student are both a BNN in our framework, we can carry out the distillation process by matching two posterior distributions $q_{SGLD}(\mathbf{w})$ and $q(\mathbf{w}|\theta)$ directly instead of matching their prediction distributions. A natural idea is considering teacher's posterior as the student's prior to regularize the variational objective in Eq (7). To make the KL term easy to compute, we assume that the $q_{SGLD}(\mathbf{w})$ is also a diagonal Gaussian distribution and then have:

$$\mathrm{KL}[q(\mathbf{w}|\theta)\|q_{SGLD}(\mathbf{w})] \approx \frac{1}{2}\sum_i\Big(\frac{\sigma_i^2 + (\mu_i - \hat{\mu}_i)^2}{\hat{\sigma}_i^2} - \log\frac{\sigma_i^2}{\hat{\sigma}_i^2} - 1\Big) \tag{10}$$

where $\hat{\mu}_i$ and $\hat{\sigma}_i$ is the mean and standard deviation of weight w_i. Such approximation about $q_{SGLD}(\mathbf{w})$ leads to computational simplicity of $\hat{\mu}_i$ and $\hat{\sigma}_i$ incremental update:

$$\begin{aligned}\hat{\mu}_i^t &\leftarrow \frac{(t-1)\hat{\mu}_i^{t-1} + w_i^t}{t} \\ \hat{\sigma}_i^t &\leftarrow \Big(\frac{(t-1)[(\hat{\sigma}_i^{t-1})^2 + (\hat{\mu}_i^{t-1} - \hat{\mu}_i^t)^2] + (w_i^t - \hat{\mu}_i^t)^2}{t}\Big)^{\frac{1}{2}}\end{aligned} \tag{11}$$

where the superscript t indexes the mini-batch iteration and $\mathbf{w}^t = \{w_i^t\}$ is the MCMC sample in current iteration. The update only depends on the two statistics in history and the last MCMC sample, therefore is suitable for online learning. We only need extra storage cost twice the number of network weights. We name this algorithm Bayesian Dark Prior Knowledge (BDPK) to distinguish it from BDK in terms of the distillation way. See Algorithm 1 for the full V-BDK and BDPK algorithms.

A baseline for BDPK is to directly approximate $q(\mathbf{w}|\theta)$ with $q_{SGLD}(\mathbf{w})$, i.e. $\mu_i = \hat{\mu}_i, \sigma_i = \hat{\sigma}_i$. In fact, Pawlowski *et al.* [2017] have exploited this idea for outlier detection. We call this method EVBE (Efficient Variational Bayesian neural network Ensemble) for short and compare it with proposed methods. Although it seems that BDPK has a similar idea with EVBE as they both compute the empirical first and second moments of MCMC posterior to approximate a variational posterior, we treat the MCMC posterior as the prior in a variational inference framework, which is essentially novel. As the variational posterior in EVBE cannot learn from data directly, when the posterior is complex and the diagonal Gaussian approximation is inaccurate, it potentially degenerates. However, BDPK only treats the diagonal Gaussian as a regularization

Algorithm 1: Two Proposed Distillation Methods

Input: training set $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$, mini-batch size M, number of iterations T, teacher learning rate η_t, student learning rate ρ_t, student optimizer **opt**, distillation methods **method**

Initialize teacher's parameters $\mathbf{w}_0$, student's parameters θ_0
if ***method*** *is "BDPK"* **then**
  Set $\{\hat{\mu}_i^0\} = 0$ and $\{\hat{\sigma}_i^0\} = 0$
for $t = 1 : T$ **do**
  // train teacher $\mathcal{T}$ (SGLD step)
  Sample a mini-batch $\mathcal{M} = \{(\mathbf{x}_{t_i}, \mathbf{y}_{t_i})\}_{i=1}^M$ from $\mathcal{D}$
  Compute $\Delta\mathbf{w}_t$ according to Eq (6) and run SGLD update: $\mathbf{w}_t := \mathbf{w}_{t-1} + \Delta\mathbf{w}_t$
  // train student $\mathcal{S}$ (Distilling step)
  if ***method*** *is "V-BDK"* **then**
    Sample a mini-batch of new input $\mathcal{M}' = \{\mathbf{x}'_{t_i}\}$ by adding noise to $\{\mathbf{x}_{t_i}\}$
    Calculate student's negative ELBO $\mathcal{L}'(\theta|\mathcal{M}', \mathbf{w}_t)$ using Eq (9) with reparameterization sampling in Eq (4) and KL term in Eq (5)
    Update student's parameters: $\theta_t := \theta_{t-1} + \mathbf{opt}(\nabla_\theta \mathcal{L}'(\theta|\mathcal{M}', \mathbf{w}_t), \rho_t)$
  if ***method*** *is "BDPK"* **then**
    Update $\{\hat{\mu}_i^t\}$ and $\{\hat{\sigma}_i^t\}$ with Eq (11)
    Calculate student's negative ELBO $\mathcal{L}(\theta|\mathcal{M})$ using Eq (7) with reparameterization sampling in Eq (4) and KL term in Eq (10)
    Update student's parameters: $\theta_t := \theta_{t-1} + \mathbf{opt}(\nabla_\theta \mathcal{L}(\theta|\mathcal{M}), \rho_t)$

and can learn from data through likelihood term, it is relatively immune to inaccurate diagonal approximation. Empirical results will show the superiority of BDPK over EVBE.

Clipping Standard Deviation of Prior

In preliminary experiments, we found the training is easy to fail due to too large KL term. In fact, according to Eq (10), this phenomenon result from some extremely small $\hat{\sigma}_i$. We avoid this problem by clipping $\hat{\sigma}_i$ with a threshold γ when calculating KL term: $\hat{\sigma}_i := \max(\hat{\sigma}_i, \gamma)$. We regard γ as a hyperparameter and tune it with validation data.

3.3 Discussion

As mentioned above, BDK [Balan *et al.*, 2015] method can hardly model epistemic uncertainty as the student is a deterministic neural network. Although the training objective allows the student to simulate the uncertainty of the predictions given by the MCMC samples in an implicit manner, it is not applicable for scenarios where epistemic uncertainty is requiredIn fact, there are some uncertainty metrics which explicitly need the epistemic uncertainty, such as predictive variance for regression [Gal and Ghahramani, 2016b] and Bayesian Active Learning by Disagreement objective (BALD) [Houlsby *et al.*, 2011] for classification. These metrics regard the disagreement between predictions given by different weight samples as uncertainty. The essential advantage of proposed methods is the ability to model epistemic uncertainty compared to BDK.

For the similar purpose with us, APD [Wang *et al.*, 2018] method also matches teacher's and student's posterior distributions but using adversarial training [Goodfellow *et al.*, 2014]. However, as stated in that paper, the number of parameters of the generator and discriminator are both 100 times that of the BNN to be distilled. In addition, due to the GAN training in each mini-batch, even the online version of APD needs to save a batch of SGLD samples. Because of the above two factors, APD method does not really solve the prohibitive storage cost problem of MCMC methods, which is the main purpose of distilling MCMC methods. In fact, the BNNs to be distilled have only about 500k parameters in Wang *et al.* [2018]. In contrast, the storage space of both two methods we propose is only a few times the number of weights of the BNN, thus our methods are much more scalable than APD.

4 Experiments

To demonstrate the effectiveness of our methods, we conduct experiments on several datasets, including MNIST, SVHN and CIFAR10. Besides reporting classification accuracy, we show a series of evaluations of uncertainty prediction, including the uncertainty on OOD datasets as well as adversarial examples, calibration results and the active learning application. The compared baselines include a DNN trained by SGD, BNNs trained by two BDL methods: SGLD & BBB, EVBE [Pawlowski *et al.*, 2017] baseline and two previous distillation methods: BDK [Balan *et al.*, 2015] & APD [Wang *et al.*, 2018]. In all experiments, we use 1 posterior sample during training and 100 posterior samples during test when BBB is employed. We use all SGLD (namely SGLD-all) samples during training to evalute predictive performance which reflects the upper bound of distillation models. Considering that saving and using all SGLD samples are not practical for downstream applications, we use 100 SGLD samples sampled from the end of training as another baseline, namely SGLD-100. Please note SGLD-100 can be regarded as a simple practical approximation of SGLD-all because it gets rid of the prohibitive storage cost. An effective distillation method should outperform it by a remarkable margin.

For MNIST, we select 10k training data for validation. We use a 2 layer MLP with 400 hidden units and ReLU activations as in previous work [Blundell *et al.*, 2015; Balan *et al.*, 2015] and treat notMNIST as OOD data following [Louizos and Welling, 2017]. This model has about 500k parameters. For SVHN and CIFAR10, we train the model on the first 5 classes (called SVHN5 and CIFAR5) and the data in the other 5 classes are considered as OOD data. We further select 10% data from training set for validation. We employ the larger LeNet architecture following Gal and Ghahramani [2016a] and Louizos and Welling [2017] and the model has about 5.74M parameters. We reimplement APD method but find that to make the model fit in an NVIDIA 1080Ti GPU with 11G memory, the hidden size can only be about 30 so that the accuracy of distilled model is always below 30%. Therefore we will not include APD for these two datasets and we argue that this phenomenon verifies the poor scalability of APD.

4.1 Predictive Performance

Table 1 shows classification accuracy of each model on three datasets. SGLD-all performs best with the ensemble predictive

Methods	MNIST		SVHN5		CIFAR5	
	Accuracy(%)	**OOD Entropy**(Bits)	**Accuracy**(%)	**OOD Entropy**(Bits)	**Accuracy**(%)	**OOD Entropy**(Bits)
SGD	98.338±0.025	0.010±0.005	92.488±0.097	0.159±0.005	73.704±0.141	0.269±0.028
BBB	97.962±0.042	1.088±0.077	93.576±0.050	0.302±0.004	77.692±0.193	0.381±0.030
SGLD-all	98.702±0.012	1.155±0.035	94.220±0.036	0.532±0.004	78.782±0.312	0.600±0.001
SGLD-100	98.292±0.049	0.335±0.030	93.660±0.035	0.260±0.004	76.576±0.111	0.273±0.004
BDK	98.254±0.019	0.296±0.005	93.338±0.071	0.241±0.016	76.784±0.204	0.301±0.004
EVBE	98.466±0.025	1.310±0.016	83.204±0.452	1.251±0.008	44.524±0.380	1.318±0.010
APD-offline*	98.232±0.016	–	–	–	–	–
APD-online	97.954±0.029	0.662±0.065	–	–	–	–
V-BDK (Proposed)	98.502±0.030	0.532±0.008	93.816±0.053	0.346±0.019	78.572±0.196	0.401±0.015
BDPK (Proposed)	98.630±0.025	0.908±0.020	93.746±0.050	0.379±0.024	78.208±0.108	0.513±0.025

Table 1: Test accuracy on three datasets and the average entropy on corresponding OOD datasets. We report the mean and standard error over 5 runs. * indicates it is not our own implementation therefore without entropy.

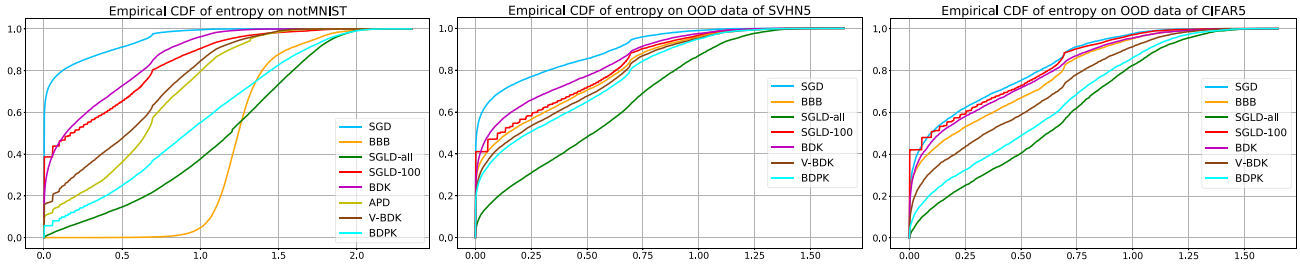

Figure 1: Empirical CDF for the entropy of the predictive distributions on three OOD datasets.

distribution from several thousand samples as expected. Both of our methods can consistently improve student BBB's performance and outperform other distillation methods, i.e. BDK and APD. Especially, sometimes our methods can achieve almost the same accuracy as SGLD-all, such as BDPK on MNIST and V-BDK on CIFAR5, with only 100 posterior samples. It is worthy noted V-BDK and BDPK are both better than SGLD-100 with the same number of posterior samples during test while BDK and APD are only comparable with or even worse than SGLD-100. For BDK, we do not have access to the original implementation and our implementation is even worse than SGD on MNIST although we try our best to tune hyperparameters. However, it indeed works better than SGD on SVHN5 and CIFAR5. For APD, with a larger thinning interval, we get a slightly better result 98.23% for offline APD than 98.1% which is reported in [Wang *et al.*, 2018] while our implementation of online APD performs a little worse. However, APD is not applicable to large network as discussed above. In summary, our methods are much more successful in achieving the purpose of distillation than BDK and APD. EVBE has a similar performance with our methods on MNIST, while it degrades sharply on the other two datasets, which attributes to the complexity of the posterior and the inaccuracies of diagonal Gaussian approximation when the DNN is large. Although for the same reason BDPK works on SVHN5 and CIFAR5 not as well as on MNIST, it can still succeed in distilling knowledge, which means it is valuable to treat the MCMC posterior as the prior of variational BNNs.

4.2 Uncertainty on OOD Datasets

We estimate the uncertainty by calculating the entropy of the predictive distributions on OOD data for each model. The ideal predictive distribution is uniform. The entropy of the predictive distribution is used to measure the uncertainty. We plot the empirical CDF of the entropies in Figure 1 following Louizos and Welling [2017] and list the average entropy over notMNIST in Table 1. CDF curves that are closer to the bottom right part of the plot are better. We plot the results of the average entropy in the middle of the five runs in Figure 1. For SGLD, we plot both SGLD-all and SGLD-100.

As expected, SGD gives the lowest entropy which confirms conventional DNNs are prone to be overconfident and not good at modeling uncertainty. SGLD-all still performs best so that it is meaningful to distill from it. Among distillation methods, BDK performs worst and is only on par with or even a little worse than SGLD-100. It seems that in terms of uncertainty modeling, BDK has little practical value. Conversely, V-BDK, BDPK perform much better than SGLD-100. We argue that it is necessary to maintain a posterior distribution for student to model uncertainty well. In particular, BDPK outperforms V-BDK by a big margin which shows this novel distillation method is promising for BDL.

It is worthy noticed that on SVHN5 and CIFAR5, BBB does not perform as well as on notMNIST and is outperformed by V-BDK and BDPK. We think such results manifest the ability of our methods to model uncertainty is transferred from SGLD, rather than the inherent property of student BBB networks.

4.3 Performance on Adversarial Examples

We also measure the robustness and predictive uncertainty of models against adversarial examples, which are produced by taking some existing data and applying a small perturbation to cause misclassification [Szegedy *et al.*, 2014]. We generate adversarial examples for each trained model using

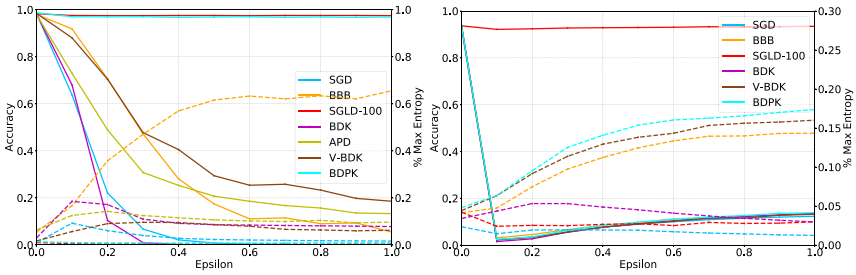

Figure 2: Accuracy (solid) vs entropy (dashed) as a function of the adversarial perturbation on MNIST (left) and SVHN5 (right).

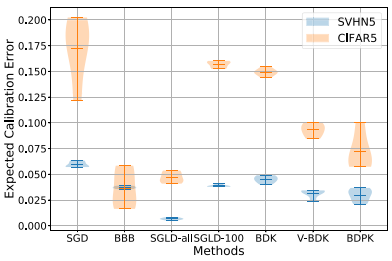

Figure 3: ECE on two datasets over 5 runs. Lower is better.

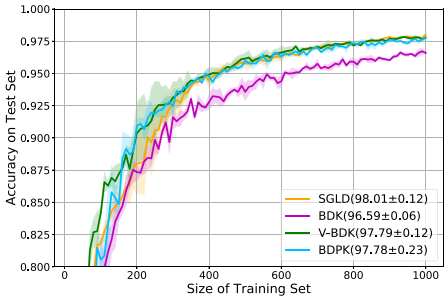

Figure 4: MNIST test accuracy as a function of number of acquired images from the pool set. The final accuracies are in parentheses.

fast gradient sign method (FGSM) [Goodfellow *et al.*, 2015] and Cleverhans library [Goodfellow *et al.*, 2016] on each test set. We plot the accuracy and percentage of average entropy relative to the maximum entropy under different magnitude of adversarial perturbation in Figure 2. Results on CIFAR5 are similiar with on SVHN5 and we do not plot due to space limit. For practical reasons, we only plot the results of SGLD-100.

On MNIST, the accuracy of each model drops rapidly apart from SGLD and BDPK. We check the gradient w.r.t input data and find there is less than 3% of the dimension of gradient non-zero in both models which causes most of the adversarial perturbation produced by FGSM to be zero. BDPK amazes us since it makes the student learn to resist the adversarial examples from SGLD successfully. As for uncertainty, only BBB predicts well. It is reasonable for SGLD and BDPK to produce extremely low entropies as the adversarial examples are almost the same as original test data.

On the other two datasets. apart from SGLD can still resist the adversarial perturbation, the performances of other models are close to 0. BDPK fails to learn to resist the adversarial examples from SGLD. We hypothesis that due to the higher dimension and complexity of data, fully factorized approximate prior is not capable of transferring this characteristic from the teacher to the student. Nevertheless, our methods are both less overconfident about the adversarial example than BDK.

4.4 Calibration Results

For classification models, calibration measures the discrepancy between prediction confidence and accuracy. A well-calibrated model can give reasonable confidence about prediction. Expected Calibration Error (ECE) is a common metric to evaluate the calibration [Guo *et al.*, 2017]. To measure the prediction uncertainty further, we report ECE on SVHN5 and CIFAR5 in Figure 3. On MNIST all methods obtain a very low ECE (<0.03) thus we do not display it due to space limit. Figure 3 shows DNN trained by SGD suffers from miscalibration while Bayesian methods, SGLD-all and BBB, are both well-calibrated. V-BDK and BDPK have much better calibration than BDK, which is on par with SGLD-100 again.

4.5 Active Learning

The performance of active learning (AL) systems heavily depends on the predictive uncertainty over unseen data. Recently BNNs have achieve good performance for AL [Gal *et al.*, 2017; Siddhant and Lipton, 2018] attributed to epistemic uncertainty. We conduct AL experiment on MNIST to demonstrate the superiority of our methods to predict uncertainty compared to BDK. For BNNs, acquisition function is selected as BALD which can exploit epistemic uncertainty and BDK uses Max Entropy as acquisition function. We select randomly a balanced initial training set of 20 images from MNIST training set and the rest of data form a pool set. At each iteration we choose 10 images from pool set with the highest predictive uncertainty given by models and add them with labels in training set. We repeat the acquisition process until the training set has 1000 images. We repeat each experiments 3 times and plot the averaged final results as well as the standard deviation in Figure 4. Our methods both outperform BDK and achieve a similar final accuracy with SGLD.

5 Conclusions

In this paper, based on the idea that distilling MCMC samples into an approximate variational posterior, we propose two novel distillation methods for variational learning of student BNNs. Our methods tackle the prohibitive storage cost problem of MCMC methods, which is the essential feature of distillation, meanwhile the student maintains the ability of teacher to model epistemic uncertainty. Compared to existing distillation methods, proposed methods have better performance on classification accuracy and predictive uncertainty.

Acknowledgments

This work is partially supported by the National High Technology Research and Development Program of China (Grant No. 2015AA015403). We would also like to thank the anonymous reviewers for their helpful comments.

References

[Balan *et al.*, 2015] Anoop Korattikara Balan, Vivek Rathod, Kevin P. Murphy, and Max Welling. Bayesian dark knowledge. In *Proceedings of NIPS*, pages 3438–3446, 2015.

[Blundell *et al.*, 2015] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *Proceedings of ICML*, pages 1613–1622, 2015.

[Chen *et al.*, 2014] Tianqi Chen, Emily B. Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *Proceedings of ICML*, pages 1683–1691, 2014.

[Depeweg *et al.*, 2017] Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft. Learning and policy search in stochastic dynamical systems with bayesian neural networks. In *Proceedings of ICLR*, 2017.

[Ding *et al.*, 2014] Nan Ding, Youhan Fang, Ryan Babbush, Changyou Chen, Robert D. Skeel, and Hartmut Neven. Bayesian sampling using stochastic gradient thermostats. In *Proceedings of NIPS*, pages 3203–3211, 2014.

[Gal and Ghahramani, 2016a] Yarin Gal and Zoubin Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference. In *ICLR, Workshop Track Proceedings*, 2016.

[Gal and Ghahramani, 2016b] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of ICML*, pages 1050–1059, 2016.

[Gal *et al.*, 2017] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *Proceedings of ICML*, pages 1183–1192, 2017.

[Gal, 2016] Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

[Goodfellow *et al.*, 2014] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of NIPS*, pages 2672–2680, 2014.

[Goodfellow *et al.*, 2015] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Proceedings of ICLR*, 2015.

[Goodfellow *et al.*, 2016] Ian J. Goodfellow, Nicolas Papernot, and Patrick D. McDaniel. cleverhans v0.1: an adversarial machine learning library. *CoRR*, abs/1610.00768, 2016.

[Graves, 2011] Alex Graves. Practical variational inference for neural networks. In *Proceedings of NIPS*, pages 2348–2356, 2011.

[Guo *et al.*, 2017] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of ICML*, pages 1321–1330, 2017.

[Hinton and van Camp, 1993] Geoffrey E. Hinton and Drew van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of COLT*, pages 5–13, 1993.

[Hinton *et al.*, 2014] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning Workshop*, 2014.

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

[Kendall and Gal, 2017] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Proceedings of NIPS*, pages 5574–5584, 2017.

[Kingma and Welling, 2014] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of ICLR*, 2014.

[Louizos and Welling, 2016] Christos Louizos and Max Welling. Structured and efficient variational deep learning with matrix gaussian posteriors. In *Proceedings of ICML*, pages 1708–1716, 2016.

[Louizos and Welling, 2017] Christos Louizos and Max Welling. Multiplicative normalizing flows for variational bayesian neural networks. In *Proceedings of ICML*, pages 2218–2227, 2017.

[MacKay, 1992] David J. C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992.

[Neal, 1995] GRadford M Neal. *Bayesian learning for neural networks*. PhD thesis, University of Toronto, 1995.

[Pawlowski *et al.*, 2017] Nick Pawlowski, Miguel Jaques, and Ben Glocker. Efficient variational bayesian neural network ensembles for outlier detection. In *ICLR, Workshop Track Proceedings*, 2017.

[Siddhant and Lipton, 2018] Aditya Siddhant and Zachary C. Lipton. Deep bayesian active learning for natural language processing: Results of a large-scale empirical study. In *Proceedings of EMNLP*, pages 2904–2909, 2018.

[Szegedy *et al.*, 2014] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *Proceedings of ICLR*, 2014.

[Wang *et al.*, 2018] Kuan-Chieh Wang, Paul Vicol, James Lucas, Li Gu, Roger Grosse, and Richard S. Zemel. Adversarial distillation of bayesian neural network posteriors. In *Proceedings of ICML*, pages 5177–5186, 2018.

[Welling and Teh, 2011] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of ICML*, pages 681–688, 2011.