DEEP LEARNING REQUIRES EXPLICIT REGULARIZATION FOR RELIABLE PREDICTIVE PROBABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

From the statistical learning perspective, complexity control via explicit regularization is a necessity for improving the generalization of over-parameterized models, which deters the memorization of intricate patterns existing only in the training data. However, the impressive generalization performance of over-parameterized neural networks with only implicit regularization challenges the importance of explicit regularization. Furthermore, explicit regularization does not prevent neural networks from memorizing unnatural patterns, such as random labels. In this work, we revisit the role and importance of explicit regularization methods for generalization of the *predictive probability*, not just the generalization of the 0-1 loss. Specifically, we analyze the possible cause of the poor predictive probability and identify that regularization of predictive confidence is required during training. We then empirically show that explicit regularization significantly improves the reliability of the predictive probability, which enables better predictive uncertainty representation and prevents the overconfidence problem. Our findings present a new direction to improve the predictive probability quality of deterministic neural networks, which can be an efficient and scalable alternative to Bayesian neural networks and ensemble methods.

1 INTRODUCTION

As deep learning models have become pervasive in real-world systems, the importance of producing a reliable predictive probability is increasing, which results in a well-calibrated behavior and a better uncertainty representation ability. The calibrated behavior refers to the ability to match its predictive probability of an event to the long-term frequency of the event occurrence (Dawid, 1982), and the uncertainty representation ability refers to the ability to represent uncertainty about its predictions. The reliable predictive probability benefits many downstream tasks such as anomaly detection (Malinin & Gales, 2019), classification with rejection (Lakshminarayanan et al., 2017), and exploration in reinforcement learning (Gal & Ghahramani, 2016). More importantly, deep learning systems with more reliable predictive probability can provide better feedback of explaining what's going on, the situations when its prediction becomes uncertain, and unexpected anomalies to users. Unfortunately, neural networks are prone to be overconfident and lack the uncertainty representation ability, and this problem becomes a fundamental concern in the deep learning community.

Bayesian methods have *inborn abilities* to produce the reliable predictive probability. Specifically, they express the probability distribution over parameters, in which uncertainty in the parameter space is automatically determined by data (MacKay, 1992; Neal, 1993). Then, uncertainty in prediction can be represented by means of providing rich information about aggregated predictions from different parameter configurations such as entropy and mutual information. In this perspective, *deterministic* neural networks selecting a single parameter configuration that cannot provide such rich information naturally lack the uncertainty representation ability. However, the automatic determination of parameter uncertainty in the light of data, i.e., *the posterior inference*, comes with prohibitive computational costs. Therefore, the mainstream of improving the predictive probability quality has been an efficient adoption of the Bayesian principle into neural networks (Gal & Ghahramani, 2016; Ritter et al., 2018; Teye et al., 2018; Joo et al., 2020a).

Recent works (Lakshminarayanan et al., 2017; Müller et al., 2019; Thulasidasan et al., 2019) discover *the hidden gems* of label smoothing (Szegedy et al., 2016), mixup (Zhang et al., 2018), and

adversarial training (Goodfellow et al., 2015), which improve the calibration performance and the uncertainty representation ability. These findings present a new possibility of improving the reliability of the predictive probability without changing the deterministic nature of neural networks. This direction is appealing because it can be applied in a plug-and-play fashion to the existing building block. This means that they can *inherit* the scalability, computational efficiency, and surprising generalization performance of the deterministic neural networks, for which the Bayesian neural networks often struggle at (Wu et al., 2019; Osawa et al., 2019; Joo et al., 2020a).

Motivated by these observations, we investigate a general direction from *the regularization perspective* to mitigate the unreliable predictive probability problem, rather than proposing new constructive heuristics or discovering hidden properties of specific methods. Our contributions can be summarized as follows: 1) we identify that the unreliable predictive probability is not caused by its deterministic nature, but rather overconfident predictions on training samples; 2) we present a new direction to alleviate the unreliable predictive behavior, which is readily applicable, computationally efficient, and scalable to large-scale models compared to Bayesian neural networks or ensemble methods; 3) our findings give a novel view on the role of explicit regularization in deep learning, which improves the reliability of the predictive probability.

2 Analyzing Cause of Unreliable Predictive Probability

We consider a classification problem with i.i.d. training samples $\mathcal{D} = \left\{(\boldsymbol{x}^{(i)}, y^{(i)})\right\}_{i=1}^{N}$ drawn from unknown distributions $P_{\mathbf{x},\mathrm{y}}$ whose corresponding tuple of random variables is $(\mathbf{x}, \mathrm{y})$. We denote $\mathcal{X}$ as an input space and $\mathcal{Y}$ as a set of categories $\{1, 2, \cdots, K\}$. Let $f^{\boldsymbol{W}} : \mathcal{X} \to \mathcal{Z}$ be a neural network with parameters $\boldsymbol{W}$ where $\mathcal{Z} = \mathbb{R}^K$ is a logit space. On top of logit space, the softmax $\sigma : \mathbb{R}^K \to \triangle^{K-1}$ normalizes exponential of logits, giving interpretation of $\sigma_k(f^{\boldsymbol{W}}(\boldsymbol{x}))$ as the predictive probability that label of $\boldsymbol{x}$ belongs to class k (Bridle, 1990):

$$\phi_k^{\boldsymbol{W}}(\boldsymbol{x}) = \frac{\exp(f_k^{\boldsymbol{W}}(\boldsymbol{x}))}{\sum_i \exp(f_i^{\boldsymbol{W}}(\boldsymbol{x}))} \tag{1}$$

where we let $\phi_k^{\boldsymbol{W}}(\boldsymbol{x}) = \sigma_k(f^{\boldsymbol{W}}(\boldsymbol{x}))$ for brevity. The probabilistic interpretation of neural network's outputs gives the natural minimization objective for classification–the cross-entropy between the predictive probability and the one-hot encoded label: $l_{CE}(\mathrm{y}, \phi^{\boldsymbol{W}}(\mathbf{x})) = -\sum_k \mathbb{1}_{\mathrm{y}}(k) \log \phi_k^{\boldsymbol{W}}(\mathbf{x})$ where $\mathbb{1}_{\mathcal{A}}(\omega)$ is an indicator function taking one if $\omega \in \mathcal{A}$ and zero otherwise. Besides, it allows extracting the predictive uncertainty of the neural network on a sample $\boldsymbol{x}$ by measuring the entropy of $\phi^{\boldsymbol{W}}(\boldsymbol{x})$, wherein the prediction with the high entropy corresponds to the answer "I don't know."

By minimizing the cross-entropy with stochastic gradient descent (SGD) (Robbins & Monro, 1951) or its variants, modern neural networks achieve the surprising generalization performance. However, the resulting neural networks are often overconfident and lack the uncertainty representation ability, which makes the interpretation of the softmax output as the "predictive probability" implausible (Gal & Ghahramani, 2016). Figure 1 illustrates the unreliable predictive behavior of ResNet (He et al., 2016): the network produces outputs with high confidence to misclassified examples (Figure 1 (upper)) and provides low predictive entropy on out-of-distribution samples, albeit the samples belong to none of the classes it learned (Figure 1 (lower)).

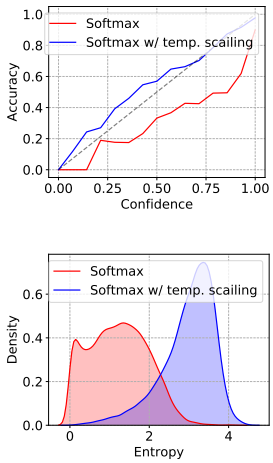
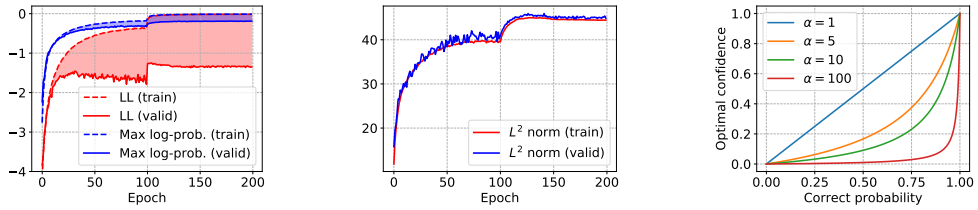


Figure 1: Reliability curve (upper) and predictive uncertainty on out-of-distribution samples (lower) of ResNet.

One simple yet effective solution to improve the predictive probability's quality is temperature scaling (Guo et al., 2017), which adjust the smoothness of the softmax so that the resulting predictive probability maximizes the log-likelihood on unseen dataset $\mathcal{D}'$:

$$\max_{\tau} \sum_{(\boldsymbol{x},y)\in\mathcal{D}'} \log \frac{\exp(f_y^{\boldsymbol{W}}(\boldsymbol{x})/\tau)}{\sum_j \exp(f_j^{\boldsymbol{W}}(\boldsymbol{x})/\tau)} \tag{2}$$

where $\boldsymbol{W}$ is a fixed pretrained weights and τ is a temperature controlling the smoothness of the softmax output. This simple method makes the softmax output more *reliable predictive probability*. For



(a) (b) (c)

Figure 2: Monitoring changes in behavior of ResNet during training on CIFAR-100. In (b), L^2 norm is approximated with respect to empirical distributions of training samples and validation samples, respectively. In (c), the optimal confidence with respect to given correct probability is described.

instance, the predictive confidence well-matches its actual accuracy, and the predictive entropy on out-of-distribution samples significantly increases (Figure 1).

Closer look at the log-likelihood on "unseen" samples. Motivated by the success of the temperature scaling, we aim to find the cause of the poor predictive probability by analyzing the log-likelihood on unseen samples and then search for alternative methods without post-tuning of temperature. This direction is appealing because it requires no extra data for temperature tuning so that it can make use of full training data. Specifically, we decompose the log-likelihood into two terms that correspond to the cases when the predictive class is equal to the label and not, respectively:

$$\begin{aligned}\mathbb{E}_{\mathbf{x},\mathrm{y}}[\log \phi^{\boldsymbol{W}}_{\mathrm{y}}(\mathbf{x})] &= \mathbb{E}_{\mathbf{x},\mathrm{y}}\left[\mathbb{1}_{\{\mathrm{y}\}}(m_{\mathbf{x}}) \log \phi^{\boldsymbol{W}}_{m_{\mathbf{x}}}(\mathbf{x}) + \sum_{k \neq m_{\mathbf{x}}} \mathbb{1}_{\{\mathrm{y}\}}(k) \log \phi^{\boldsymbol{W}}_{k}(\mathbf{x})\right] \\ &\leq \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{\mathrm{y}|\mathbf{x}}\left[\mathbb{1}_{\{\mathrm{y}\}}(m_{\mathbf{x}})\right] \log \phi^{\boldsymbol{W}}_{m_{\mathbf{x}}}(\mathbf{x}) + \left(1 - \mathbb{E}_{\mathrm{y}|\mathbf{x}}\left[\mathbb{1}_{\{\mathrm{y}\}}(m_{\mathbf{x}})\right]\right) \log\left(1 - \phi^{\boldsymbol{W}}_{m_{\mathbf{x}}}(\mathbf{x})\right)\right] \quad (3)\end{aligned}$$

where $m_{\mathbf{x}}$ is the predictive class such that $m_{\mathbf{x}} = \arg\max_k f^{\boldsymbol{W}}_k(\mathbf{x})$ and the inequality accounts for dispersion of non-maximum probability into $K-1$ categories. Therefore, given a sample $\boldsymbol{x} \sim \mathbf{x}$, the upper bound of the log-likelihood is determined by the realization of a "stochastic switch" $\mathbb{E}_{\mathrm{y}|\boldsymbol{x}}\left[\mathbb{1}_{\{\mathrm{y}\}}(m_{\boldsymbol{x}})\right]$ that selects between two deterministic scores $\log \phi^{\boldsymbol{W}}_{m_{\boldsymbol{x}}}(\boldsymbol{x})$ and $\log(1 - \phi^{\boldsymbol{W}}_{m_{\boldsymbol{x}}}(\boldsymbol{x}))$. Here, we note the impact of the implicit bias of the cross-entropy minimization on these scores. Specifically, the minimum of the cross-entropy is unattainable so that SGD updates $\boldsymbol{W}$ in the direction that increases $\phi^{\boldsymbol{W}}_{m_{\boldsymbol{x}}}(\boldsymbol{x})$ and decreases $\phi^{\boldsymbol{W}}_{k}(\boldsymbol{x}), \forall k \neq m_{\boldsymbol{x}}$ whenever it corrects the answer. Considering modern neural networks easily fit most of samples even in the early stage of training, repetitive training makes the score $\log \phi^{\boldsymbol{W}}_{m_{\boldsymbol{x}}}(\boldsymbol{x})$ near zero and the score $\log(1-\phi^{\boldsymbol{W}}_{m_{\boldsymbol{x}}}(\boldsymbol{x}))$ significantly small; as a result, the neural network becomes vulnerable to the severe penalty for misclassified samples. Figure 2 (a) show that the maximum entries of log probability on both training and validation samples are consistently increasing during training and eventually getting close to zero.

However, the desired level of the predictive confidence is significantly smaller than what it used to be. To derive the optimal confidence level, we first transform the inequality in the equation 3 to the equality. Fortunately, there exists a divisor $\alpha_{\mathbf{x},\mathrm{y}} \geq 1$ to $(1-\phi^{\boldsymbol{W}}_{m_{\mathbf{x}}}(\mathbf{x}))$ in the second term of the upper bound, which gives the equality. Here, $\alpha_{\mathbf{x},\mathrm{y}}$ represents the ratio of predictive probability for the label to $(1 - \phi^{\boldsymbol{W}}_{m_{\mathbf{x}}}(\mathbf{x}))$, in which smaller value of $\alpha_{\mathbf{x},\mathrm{y}}$ indicates more concentration of $1 - \phi^{\boldsymbol{W}}_{m_{\mathbf{x}}}(\mathbf{x})$ on the true label position and $\alpha_{\mathbf{x},\mathbf{y}} = 1$ means the neural network achieves the perfect top-2 accuracy. Then, we can derive the optimal confidence level at $\mathbf{x}$ given *fixed test accuracy* and $\alpha_{\mathbf{x},\mathbf{y}}$ by using the concavity of $\mathbb{E}_{\mathrm{y}}[\log \phi^{\boldsymbol{W}}_{\mathrm{y}}(\mathbf{x})]$: $\phi^{\boldsymbol{W}^*}_{m_{\mathbf{x}}}(\mathbf{x}) = \mathbb{E}_{\mathrm{y}|\mathbf{x}}[\mathbb{1}_{\{\mathrm{y}\}}(m_{\mathbf{x}})] / \left(\mathbb{E}_{\mathrm{y}|\mathbf{x}}[\mathbb{1}_{\{\mathrm{y}\}}(m_{\mathbf{x}})](1 - \alpha_{\mathbf{x},\mathbf{y}}) + \alpha_{\mathbf{x},\mathbf{y}}\right)$. Figure 2 (c) illustrates the optimal confidence with respect to $\mathbb{E}_{\mathrm{y}|\mathbf{x}}[\mathbb{1}_{\{\mathrm{y}\}}(m_{\mathbf{x}})]$ for various values of $\alpha_{\mathbf{x},\mathbf{y}}$. The optimal confidence level at $\mathbf{x}$ is equal to its expected accuracy at $\mathbf{x}$ when $\alpha_{\mathbf{x},\mathbf{y}} = 1$ and smaller than the accuracy in all other cases when $\alpha_{\mathbf{x},\mathbf{y}} > 1$. However, the current level of predictive confidence is significantly greater than the desired value. For example, in Figure 2 (a), the predictive confidence is very close to 1 but its accuracy is around 75%. Therefore, bridging this gap by improving accuracy or reducing the confidence is required to improve the log-likelihood.

Implications. The above analysis suggests useful implications about the desirability of the confidence control for improving the log-likelihood on unseen samples. First, it shows that optimal confidence is increasing rapidly around the perfect accuracy when $\alpha_{\mathbf{x},\mathbf{y}} > 1$ (Figure 2 (c)). There-

fore, if test accuracy is not significantly high enough, improving accuracy by a moderate margin will not dramatically impact the optimal confidence level and consequently may not result in the log-likelihood improvement. Furthermore, improving $\mathbb{E}_{\mathrm{y}|\mathbf{x}}[\mathbb{1}_{\{\mathrm{y}\}}(m_{\mathbf{x}})]$ on unseen samples by manipulating its value on training samples is extremely challenging. Specifically, $\mathbb{E}_{\mathrm{y}|\mathbf{x}}[\mathbb{1}_{\{\mathrm{y}\}}(m_{\mathbf{x}})]$ depends on the external randomness $\mathrm{y}|\mathbf{x}$ and many complex factors, such as input dimensionality, model complexity, and inherent noise (Vapnik, 1995; Bartlett et al., 2006). Therefore, improving $\mathbb{E}_{\mathrm{y}|\mathbf{x}}[\mathbb{1}_{\{\mathrm{y}\}}(m_{\mathbf{x}})]$ on training samples may not be directly connected to its improvement on unseen samples. Conversely, the predictive confidence $\phi^{\boldsymbol{W}}_{m_{\mathbf{x}}}(\mathbf{x})$ is the property of $f^{\boldsymbol{W}}$ itself. This means that the difference between its value on training samples and unseen samples would be mostly caused by the variance of the Monte-Carlo estimation with a finite sample size. That is, restricting the predictive confidence $\phi^{\boldsymbol{W}}_{m_{\mathbf{x}}}(\mathbf{x})$ on training samples will directly impact the predictive confidence on unseen samples and therefore can improve the reliability of the predictive probability. For example, the empirical mean of the maximum log-probability $\log \phi^{\boldsymbol{W}}_{m_{\mathbf{x}}}(\mathbf{x})$ (Figure 2 (a)) and L^2 norm of $f^{\boldsymbol{W}}$ (Figure 2 (b)), which are properties of $f^{\boldsymbol{W}}$, on training samples are significantly similar to those values on unseen samples compared to the log-likelihood (Figure 2 (a)), which depends on the external randomness. For these reasons, we concentrate on the direction of reducing predictive confidence during training in order to improve the reliability of the predictive probability for the rest of the paper.

3 Confidence control for reliable predictive probability

The goal of this section is to examine the effectiveness of restricting predictive confidence during training on improving the reliability of predictive probability. To this end, we apply several existing regularizers to the (pre-activation) ResNet (He et al., 2016) trained on CIFAR (Krizhevsky et al., 2009), which is one of the most prevalent basis models in many state-of-the-art architectures (Huang et al., 2017; Xie et al., 2017). We also present the VGG (Simonyan & Zisserman, 2015) as a representative of models without residual connection in appendix B, in which we observe results similar to ResNet. We follow the standard training strategy presented in He et al. (2016) except initial learning rate warm-up, clipping gradient when its norm exceeds one, and making extra validation set of 10,000 samples split from the training set. We describe a detailed setup in appendix A.

To measure the reliability of the predictive probability, we employ the following metrics commonly used in literature (Lakshminarayanan et al., 2017; Ovadia et al., 2019; Guo et al., 2017):

- Negative log-likelihood (NLL) evaluates how well the predictive probability explains the test data $\mathcal{D}^T$, whose optimal score is achieved if and only if $\phi^{\boldsymbol{W}}(\mathbf{x})$ perfectly match $p(\mathrm{y}|\mathbf{x})$.
- Expected calibration error (ECE) (Naeini et al., 2015) evaluates how well the predictive confidence matches its actual accuracy. Specifically, ECE on $\mathcal{D}^T$ is computed by binning predictions into M-groups based on their confidences such that $\mathcal{G}_i = \left\{\boldsymbol{x} : i/M < \max_k \phi^{\boldsymbol{W}}_k(\boldsymbol{x}) \leq (1+i)/M, \boldsymbol{x} \in \mathcal{D}^T\right\}$, then averaging their calibration scores by $\sum_{i=1}^{M} \frac{|\mathcal{G}_i|}{|\mathcal{D}^T|} |\mathrm{acc}(\mathcal{G}_i) - \mathrm{conf}(\mathcal{G}_i)|$ where $\mathrm{acc}(\mathcal{G}_i)$ and $\mathrm{conf}(\mathcal{G}_i)$ are average accuracy and confidence of predictions in group $\mathcal{G}_i$, respectively.
- Predictive entropy on out-of-distribution (OOD) samples evaluates how well the predictive uncertainty represents their ignorance. Here, the reliable predictions ought to produce predictions with high entropy since the samples belong to none-of-classes seen during training.

3.1 Confidence control by weight decay

The simplest way to constrain the confidence may conceivably be using stronger weight decay (Krogh & Hertz, 1992) that can encourage to produce less extreme outputs by shrinking weights. Therefore, we first explore the confidence control by varying the weight decay ratio λ, conjecturing that the weight decay ratio used in the base model, e.g., $\lambda = 0.0001$ in ResNet, is too small to prevent overconfident predictions. Figure 3 (upper) illustrates the impact of λ on generalization performance and calibration performance. We can see that ECE decreases to some extent, as the decay ratio increases. However, the ECE improvement becomes *inversely* proportional to a generalization performance improvement when the decay ratio is larger than 0.001. This means that improving the reliability with strong weight decay is against the primary goal of supervised learning.

We further investigate this undesirability by monitoring L^2 function norm of $f^{\boldsymbol{W}}$ (cf. equation 5) under different weight decay ratios. This analysis is motivated by the fact that temperature scaling modifies only L^p norm of $f^{\boldsymbol{W}}$ as it divides logits with a scalar. Under the single-label classification model, this measure is a useful proxy for the predominance of the maximum predictive probability. For setting the desired value of L^2 norm, we "leak" the test set $\mathcal{D}^T$ then apply the temperature scaling to the neural network trained with $\lambda = 0.0001$.

Figure 3 shows that the SGD with various decay ratios finds only *the trivial solution* or *an infeasible solution* to the following optimization problem:

$$\min_{\boldsymbol{W}} \hat{\mathbb{E}}_{\mathcal{D}} \left[l_{CE}(\mathrm{y}, \phi^{\boldsymbol{W}}(\mathbf{x})) \right] \quad \text{s.t.} \parallel f^{\boldsymbol{W}} \parallel_2 \leq \parallel f^{\boldsymbol{W}'}/\tau^* \parallel_2 \tag{4}$$

where $\hat{\mathbb{E}}_{\mathcal{D}}[\cdot]$ is the empirical mean on $\mathcal{D}$, $\boldsymbol{W}'$ is the trained weights under $\lambda = 0.0001$, and $\tau^* = \arg\max_\tau \hat{\mathbb{E}}_{\mathcal{D}^T} \left[\log(\phi_{\mathrm{y}}^{\boldsymbol{W}'}(\mathbf{x})/\tau) \right]$. This means that reducing the predictive confidence to the desired level without sacrificing generalization performances is a significantly challenging optimization problem. Specifically, Figure 3 (lower) shows that the L^2 norm goes to zero under the decay ratio $\lambda \geq 0.003$, which means that all weights collapse to zero, i.e., the trivial solution. This happens when the magnitude of decay ratio overwhelms the magnitude of weight update by the gradient of the cross-entropy, e.g., around at the epoch 50 under $\lambda = 0.003$ (Figure 3 (lower)). On the other hand, ratios of 0.001 and 0.0001 do not suffer from the weight collapse, but a scale of L^2 norm under such ratios is significantly higher than $\parallel f^{\boldsymbol{W}'}/\tau^* \parallel_2$, which correspond to infeasible solutions (Figure 3 (lower)). These results may seem natural because the weight decay does not explicitly consider the constraint about the predictive confidence. Therefore, we explore an alternative way that adds a regularization loss explicitly concerning the predictive confidence, e.g., the constraint in equation 4.

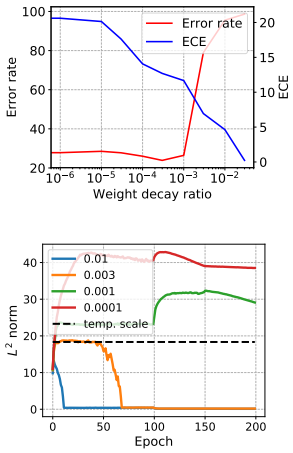


Figure 3: Impact of the weight decay ratio on ECE and accuracy (upper) and L^2 norm (lower).

3.2 Direct confidence control by explicit regularization

In this subsection, we examine two types of regularizers that directly constrain the predictive confidence on training samples.

Regularization in the function space. The first approach regards $f^{\boldsymbol{W}}$ as an element of $L^p(\mathcal{X})$ space. L^p space is the space of measurable functions with the norm:

$$\parallel f^{\boldsymbol{W}} \parallel_p = \left(\int_{\mathcal{X}} |f^{\boldsymbol{W}}(\boldsymbol{x})|^p dP_{\mathbf{x}}(\boldsymbol{x}) \right)^{1/p} < \infty \tag{5}$$

Here, we note that the norm is computed with respect to the input generating distribution $P_{\mathbf{x}}$, which allows concerning how the function $f^{\boldsymbol{W}}$ actually behaves on the data manifold. Since $P_{\mathbf{x}}$ is unknown, so it is approximated by the Monte-Carlo approximation with mini-batch samples. Then, the approximate function norm can be computed by $\parallel f^{\boldsymbol{W}} \parallel_p^p \approx \frac{1}{m} \sum_{i,j} |f_j^{\boldsymbol{W}}(\boldsymbol{x}^{(i)})|^p$. By penalizing the complexity with the L^p norm, the continuous increase in the leading entry of logit towards infinity, or continuous decreases in the non-leading entry to negative infinity, can be prevented. In this paper, we examine $\parallel f^{\boldsymbol{W}} \parallel_1$ and $\parallel f^{\boldsymbol{W}} \parallel_2^2$ regularization losses.

Regularization in the probability distribution space. The second approach regards $f^{\boldsymbol{W}}(\mathbf{x})$ as a random variable, then minimize its distance to the bell-shaped target distribution. Since the bell-shaped distribution has peak mass at zero and the same density on points with the same distance to zero, this regularization can make logits have small values and make each entry of logits have a similar scale, which can prevent neural nets from producing overconfident answers. Among various bell-shaped distributions, we chose the standard Gaussian for the target distribution due to its simplicity and popularity. As a metric in the probability distribution space, we use the sliced Wasserstein distance of order one because of its computational efficiency and ability to measure the distance between probability distributions with different supports, which is useful when dealing with the empirical distribution. We refer to Peyré et al. (2019) for more detailed explanations

Table 1: Experimental results under various regularization methods. Arrows on the metrics represent the desirable direction. We searched four hyperparameters for each method and choose the best hyperparameter based on validation accuracy (cf. appendix A). Values represent $\mu \pm \sigma$ obtained from five repetitions, and all values are rounded to two decimal places.

Model & Data	Regularizer	Acc ↑	NLL ↓	ECE ↓	$\| f^{\boldsymbol{W}} \|_2$
ResNet-50 & CIFAR-10	Vanilla	94.17 ±0.08	0.29 ±0.02	4.06 ±0.17	20.12 ±0.1
	$\| f^{\boldsymbol{W}} \|_1$	94.32 ±0.13	0.25 ±0.01	2.89 ±0.08	6.81 ±0.01
	$\| f^{\boldsymbol{W}} \|_2^2$	94.38 ±0.13	0.23 ±0.01	3.27 ±0.14	8.19 ±0.03
	$SW_1(\mu^{\boldsymbol{W}}_{\mathcal{D}'}, \nu)$	94.46 ±0.06	0.23 ±0.0	2.12 ±0.19	5.4 ±0.02
	PER	94.30 ±0.14	0.23 ±0.0	2.89 ±0.13	6.94 ±0.05
ResNet50 & CIFAR-100	Vanilla	74.64 ±0.04	1.31 ±0.02	13.95 ±0.26	43.81 ±0.08
	$\| f^{\boldsymbol{W}} \|_1$	76.28 ±0.50	1.27 ±0.02	7.77 ±0.24	9.07 ±0.07
	$\| f^{\boldsymbol{W}} \|_2^2$	75.84 ±0.53	1.07 ±0.02	5.52 ±0.31	10.55 ±0.08
	$SW_1(\mu^{\boldsymbol{W}}_{\mathcal{D}'}, \nu)$	76.27 ±0.33	1.1 ±0.01	7.02 ±0.31	9.67 ±0.06
	PER	76.23 ±0.28	1.15 ±0.0	4.67 ±0.3	7.56 ±0.11

about this metric. Specifically, given mini-batch samples $\mathcal{D}' = \{\boldsymbol{x}^{(i)}\}_{i=1}^m$, let an empirical measure of logits be $\mu^{\boldsymbol{W}}_{\mathcal{D}'}(\mathbb{A}) = \frac{1}{m}\sum_i \mathbb{1}_{\mathbb{A}}(f^{\boldsymbol{W}}(\boldsymbol{x}^{(i)}))$ and the standard Gaussian measure on $\mathcal{Z}$ be $\nu(\mathbb{A}) = \frac{1}{2\pi^{K/2}}\int_{\mathbb{A}} \exp\left(-\frac{1}{2} \| \boldsymbol{z} \|^2\right) d\boldsymbol{z}$. Then, the sliced Wasserstein distance can be computed by:

$$SW_1(\mu^{\boldsymbol{W}}_{\mathcal{D}'}, \nu) = \int_{\mathbb{S}^{K-1}} \int_{-\infty}^{\infty} \left| F_{\mu_\theta}(x) - \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}_{(\infty,x)}(\langle \boldsymbol{z}^{(i)}, \theta \rangle) \right| dx d\lambda(\theta) \tag{6}$$

where $\boldsymbol{z}^{(i)} = f^{\boldsymbol{W}}(\boldsymbol{x}^{(i)})$, λ is a uniform measure on the unit sphere $\mathbb{S}^{K-1}$, and μ_θ is a measure obtained by projecting $\mu^{\boldsymbol{W}}_{\mathcal{D}'}$ at angle θ. Projected error function regularization (PER) (Joo et al., 2020b) simplifies the computation of the $SW_1(\mu^{\boldsymbol{W}}_{\mathcal{D}'}, \nu)$ by applying Minkowski inequality to the above equation. As a result, the gradient of PER resembles the gradient of Huber loss (Huber, 1964) in the projected space, which allows combining advantages of both L^1 norm and L^2 norm as well as capturing dependency between logits of each location by a projection operation (Joo et al., 2020b).

Results. Table 1 lists the experimental results, in which both regularization in the function space and the Wasserstein probability space successfully control the confidence, e.g., reducing L^2 norm of ResNet at least 34% on CIFAR-10 and 68% CIFAR-100. We note that regularization methods can constrain the confidence without compromising the generalization performance; actually, all regularization methods give small but consistent improvements of test error rates. We also remark that the sum of the Frobenius norm of weights often increases compared to the vanilla method and changes only at most 2% when it decreases, which again shows the undesirability of adjusting the weight decay ratio for confidence control.

More importantly, the predictive probability's reliability significantly improves under all considered measures compared to the vanilla method. For instance, the regularization methods reduce NLL of ResNet at least 13% CIFAR-10 and 6% on CIFAR-100 and reduce ECE of ResNet at least 19% on CIFAR-10 and 41% on CIFAR-100. These improvements are comparable to or better than those of temperature scaling. For instance, ResNet with temperature scaling gives NLL of 1.15 and ECE of 8.41 on CIFAR-100. Here, we split the test set into two equal-size sets–a performance measurement set and a temperature calibration set–and measure the performance after temperature scaling with the calibration set, and repeat the same procedure by reversing their roles. We want to remark that the more realistic evaluation requires to draw the temperature calibration set from the training set. In this case, its performance would decrease as it cannot fully exploit the entire dataset during training.

We also investigate the usefulness of explicit regularization on the uncertainty representation ability on OOD samples. Since OOD samples don't belong to any categories, the neural network should produce the answer of "I don't know." Figure 4 illustrates predictive uncertainty of ResNet-50 with respect to CIFAR-100 (in-distribution) and SVHN (Netzer et al., 2011) (OOD). Vanilla method's predictive uncertainty on SVHN remains in the somewhat confident region, albeit less confident compared to those on CIFAR-100. On the contrary, explicit regularization successfully gathers a mass of predictive uncertainty for SVHN samples around the maximum-entropy region.

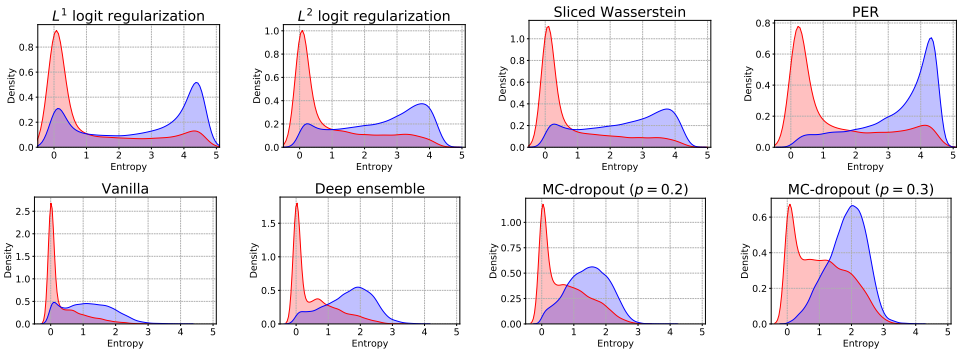
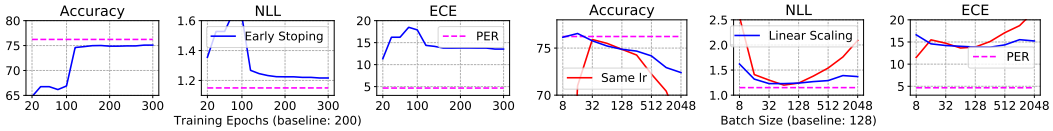


Figure 4: Density of predictive uncertainty on CIFAR-100 (in-distribution) and SVHN (OOD). Upper figures illustrate explicit regularization methods and lower figures illustrate vanilla method, ensemble methods, and Bayesian neural networks.



(a) Early stopping (b) Varying noise of stochastic gradient

Figure 5: Effects of implicit regularization methods on accuracy, negative log-likelihood, and ECE.

We compare the uncertainty representation abilities of regularization methods to Bayesian neural networks and ensemble methods. Specifically, we use the scalable Bayesian neural network, called MC-dropout (Gal & Ghahramani, 2016), because other methods based variational inference (Graves, 2011; Blundell et al., 2015; Wu et al., 2019) or MCMC (Welling & Teh, 2011; Zhang et al., 2020) requires modifications to the baseline including the optimization procedure and the architecture, which deters a fair comparison. We searched a dropout rate over $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ and use 100 number of Monte-Carlo samples at test time, i.e., 100x more inference time. We also use the deep ensemble (Lakshminarayanan et al., 2017) with 5 number of ensembles, i.e., 5x more training and inference time. Figure 4 shows that the regularization-based methods produce significantly better uncertainty representation than the MC-dropout and deep ensemble; even though both deep ensemble and MC-dropout have the ability to move mass on less certain regions, the positions are still far from the highest uncertainty region, unlike the regularization-based methods. Finally, we note the several approaches (Malinin & Gales, 2019; Papadopoulos et al., 2019) explicitly maximizing uncertainty on OOD training samples for OOD detection tasks. However, we do not consider such approaches, assuming we have no access to OOD samples during training.

3.3 Can the improved reliability be attributed to improved generalization?

Considering the log-likelihood is the training objective, the improved reliability may be caused by the improved generalization *due to regularization effects*. To clarify this point, we investigate the impact of improved generalization via implicit regularization mechanisms (early stopping and exploitation of noise from SGD) to ResNet-50 on CIFAR-100. Figure 5 (a) shows that the early stopping does not result in dramatic improvements in all measures compared to PER. Instead, NLL and ECE slightly improve under longer training due to increased classification accuracy. Considering the L^2 norm of logits achieves a significantly high value at the early stage of training (cf. Figure 2 (b)), this undesirable result may seem natural. Next, we examine the effects of varying batch sizes under baseline learning rate or linear scaling rule (Goyal et al., 2017), in which a smaller batch size increases the noise of the stochastic gradient. In Figure 5 (b), we observe that increasing noise level by using smaller batch sizes benefits generalization performance. However, it does not benefit and even worsens the predictive distribution quality (NLL and ECE). Besides, we observed that these implicit regularizers fail to produce high predictive entropy on OOD samples, unlike explicit regularization. These observations corroborate the efficacy of confidence control on improving the reliability of predictive probability, which highlights the importance of explicit regularization.

4 RELATED WORK

Recent works show that joint modeling of a generative model $p(\mathbf{x})$ along with a classifier $p(\mathrm{y}|\mathbf{x})$, or $p(\mathbf{x}, \mathrm{y})$ directly, helps to produce the reliable predictive probability (Alemi et al., 2018; Nalisnick et al., 2019; Grathwohl et al., 2020). Specifically, Alemi et al. (2018) argue that modeling stochastic hidden representation through variational information bottleneck principle (Alemi et al., 2017) allows representing better predictive uncertainty. This can be related to the effectiveness of ensemble methods, which aggregate representations of several models, on enhancing predictive uncertainty representation and confidence calibration (Lakshminarayanan et al., 2017; Ovadia et al., 2019; Ashukha et al., 2020). In this regard, hybrid modeling and ensemble methods share a similar principle to the Bayesian methods, concerning the *stochasticity of the function*. However, this paper concentrates on explicit regularization for controlling the predictive confidence, which is fundamentally different from previous works focusing on building the stochastic representation.

Other works concentrate on *structural characteristics* of the neural networks. Specifically, Hein et al. (2019) identify the cause of the overconfidence problem based on an analysis of the affine compositional function, e.g., ReLU. The basic intuition behind this analysis is that one can always find a multiplier λ to an input $\boldsymbol{x}$, which makes a neural network to produce one dominant entry on $\lambda\boldsymbol{x}$. Verma & Swami (2019) point out that the region of the highest predictive uncertainty under the softmax forms a subspace in the logit space, so the volume of an area of representing high predictive uncertainty would be negligible. However, our approach suggests that these structural characteristics' inherent flaws can be easily cured by adding an explicit regularization term without changing the existing components of neural networks.

From the perspective of the statistical learning theory (Vapnik, 1995), a regularization method minimizing some form of complexity measures, e.g., Rademacher complexity (Bartlett et al., 2005) or VC-dimension (Vapnik & Chervonenkis, 2015), is a "must" to achieve better generalization of over-parameterized models by preventing memorization of intricate patterns existing only in training samples. However, the role of capacity control with explicit regularization is challenged by many observations in deep learning. Specifically, over-parameterized neural networks achieve impressive generalization performance with only implicit regularizations contained in the optimization procedures (Hardt et al., 2016; Li et al., 2019) or the structural characteristics (Gunasekar et al., 2018; Hanin & Rolnick, 2019; Luo et al., 2019). Even more, Zhang et al. (2017) show that the explicit regularization cannot prevent neural networks from easily fitting random labels that *cannot be generalized* to unseen examples. Therefore, the importance of capacity control with explicit regularization seems to be questionable in deep learning. In this work, we re-emphasize its importance, presenting a different view on the role of regularization in terms of *generalization of the predictive probability*, not solely on better accuracy.

5 CONCLUSION

In this works, we show that *"deep learning requires explicit regularization for reliable predictive probability,"* presenting a novel view on the role and importance of explicit regularization. Specifically, we systematically analyze factors contributing to the unreliable predictive probability of deep neural networks by decomposing the log-likelihood into the *stochastic switch*, i.e., whether the predictive class matches the label, that chooses scores between two *deterministic scores*–log of maximum predictive confidence and log of a part of the remaining predictive confidence. We then show the effectiveness of explicit regularization on improving the reliability of the predictive probability, which in turn improves calibration, uncertainty representation, and even the test accuracy. This direction is appealing in terms of computational efficiency and scalability compared to the Bayesian and ensemble methods. Despite these advantages, the regularization methods are limited in that they cannot utilize more sophisticated uncertainty quantification methods based on stochastic representation on the predictive probability space, such as mutual information measuring epistemic uncertainty (Smith & Gal, 2018), due to its deterministic nature. We leave this limitation as an important future direction of research, which may be solved by more expressive parameterization, e.g., (Wilson et al., 2016; Skafte et al., 2019; Malinin & Gales, 2019; Joo et al., 2020a).

REFERENCES

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017.

Alexander A Alemi, Ian Fischer, and Joshua V Dillon. Uncertainty in the variational information bottleneck. *arXiv preprint arXiv:1807.00906*, 2018.

Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. In *International Conference on Learning Representations*, 2020.

Peter L Bartlett, Olivier Bousquet, Shahar Mendelson, et al. Local Rademacher complexities. *The Annals of Statistics*, 33(4):1497–1537, 2005.

Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *International Conference on Machine Learning*, 2015.

John S Bridle. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing*, pp. 227–236. Springer, 1990.

A Philip Dawid. The well-calibrated Bayesian. *Journal of the American Statistical Association*, 77 (379):605–610, 1982.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, 2016.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2020.

Alex Graves. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems*, 2011.

Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. In *Advances in Neural Information Processing Systems*, 2018.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.

Boris Hanin and David Rolnick. Deep relu networks have surprisingly few activation patterns. In *Advances in Neural Information Processing Systems*, 2019.

Moritz Hardt, Benjamin Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *IEEE International Conference on Computer Vision*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pp. 630–645, 2016.

Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Peter J Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35 (1):73–101, 1964.

Taejong Joo, Uijung Chung, and Min-Gwan Seo. Being Bayesian about categorical probability. In *International Conference on Machine Learning*, 2020a.

Taejong Joo, Donggu Kang, and Byunghoon Kim. Regularizing activations in neural networks via distribution matching with the Wasserstein metric. In *International Conference on Learning Representations*, 2020b.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Anders Krogh and John A Hertz. A simple weight decay can improve generalization. In *Advances in Neural Information Processing Systems*, 1992.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, 2017.

Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. In *Advances in Neural Information Processing Systems*, 2019.

Ping Luo, Xinjiang Wang, Wenqi Shao, and Zhanglin Peng. Towards understanding regularization in batch normalization. In *International Conference on Learning Representations*, 2019.

David JC MacKay. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992.

Andrey Malinin and Mark Gales. Reverse KL-divergence training of prior networks: Improved uncertainty and adversarial robustness. In *Advances in Neural Information Processing Systems*, 2019.

Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems*, 2019.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using Bayesian binning. In *AAAI Conference on Artificial Intelligence*, 2015.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Hybrid models with deep and invertible features. In *International Conference on Machine Learning*, 2019.

Radford M Neal. Bayesian learning via stochastic dynamics. In *Advances in Neural Information Processing Systems*, 1993.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Kazuki Osawa, Siddharth Swaroop, Mohammad Emtiyaz E Khan, Anirudh Jain, Runa Eschenhagen, Richard E Turner, and Rio Yokota. Practical deep learning with Bayesian principles. In *Advances in Neural Information Processing Systems*, 2019.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, 2019.

Aristotelis-Angelos Papadopoulos, Mohammad Reza Rajati, Nazim Shaikh, and Jiamian Wang. Outlier exposure with confidence control for out-of-distribution detection. *arXiv preprint arXiv:1906.03509*, 2019.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable Laplace approximation for neural networks. In *International Conference on Learning Representations*, 2018.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Nicki Skafte, Martin Jørgensen, and Søren Hauberg. Reliable training and estimation of variance networks. In *Advances in Neural Information Processing Systems*, 2019.

Lewis Smith and Yarin Gal. Understanding measures of uncertainty for adversarial example detection. *arXiv preprint arXiv:1803.08533*, 2018.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Mattias Teye, Hossein Azizpour, and Kevin Smith. Bayesian uncertainty estimation for batch normalized deep networks. In *International Conference on Machine Learning*, 2018.

Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *Advances in Neural Information Processing Systems*, 2019.

Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

Vladimir N Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of Complexity*, pp. 11–30. Springer, 2015.

Gunjan Verma and Ananthram Swami. Error correcting output codes improve probability estimation and adversarial robustness of deep neural networks. In *Advances in Neural Information Processing Systems*, 2019.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *International Conference on Machine Learning*, 2011.

Andrew G Wilson, Zhiting Hu, Ruslan R Salakhutdinov, and Eric P Xing. Deep kernel learning. In *International Conference on Artificial Intelligence and Statistics*, 2016.

Anqi Wu, Sebastian Nowozin, Edward Meeds, Richard E Turner, José Miguel Hernández-Lobato, and Alexander L Gaunt. Deterministic variational inference for robust Bayesian neural networks. In *International Conference on Learning Representations*, 2019.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical stochastic gradient MCMC for Bayesian deep learning. In *International Conference on Learning Representations*, 2020.

A DETAILED EXPERIMENTAL SETUP

ResNet base setup. We trained ResNet for 200 epochs by SGD with momentum coefficient 0.9, mini-batch size of 128, and a weight decay ratio 0.0001; weights were initialized by He initialization (He et al., 2015); an initial learning rate was 0.1, and decreased by a factor of 10 at 100 and 150 epochs; image pixel values were subtracted by the mean and divided by the standard deviation, zero-padded with 4 pixels, randomly cropped to 32x32, and horizontally flipped with the probability of 0.5.

VGG setup. We trained VGG by re-using the ResNet setup for convenience, except increasing the weight decay ratio to 0.0005 as in (Simonyan & Zisserman, 2015).

Hyperparameters. We searched four regularization loss coefficients for each method, and chose best one based on validation set accuracy (Table 2). The search spaces were: $\{0.1, 0.03, 0.01, 0.003\}$ for L^1 norm; $\{0.03, 0.01, 0.003, 0.001\}$ for L^2 norm; $\{0.1, 0.03, 0.01, 0.003\}$ for sliced Wasserstein regularization; $\{1.0, 0.3, 0.1, 0.03\}$ for PER (10x lower coefficient for CIFAR-10).

Sliced Wasserstein regularization and PER involve the integral over the unit sphere, which is evaluated by Monte-Carlo approximation. In this paper, we used 256 number of evaluations, following (Joo et al., 2020b).

Table 2: Best hyperparameters for each configuration

Regularizer	VGG-16 & CIFAR-10	VGG-16 & CIFAR-100	ResNet-50 & CIFAR-10	ResNet-50 & CIFAR-100
$\| f^{\boldsymbol{W}} \|_1$	0.01	0.03	0.01	0.01
$\| f^{\boldsymbol{W}} \|_2^2$	0.003	0.01	0.003	0.01
$SW_1(\mu^{\boldsymbol{W}}_{\mathcal{D}'}, \nu)$	0.001	0.03	0.001	0.01
PER	0.003	1.0	0.03	1.0

B VGG RESULTS

As consistent with the results of ResNet, all regularization losses improves NLL, ECE, and accuracy (Table 3), except L^1 regularization on CIFAR-100. However, the improvements are less significant compared to ResNet because the small capacity of VGG makes the vanilla method produces less confident answers and then less vulnerable to the confidence penalty. This can be inferred from that values of $\| f^{\boldsymbol{W}} \|_2$ of VGG are reduced by almost 50% compared to those of ResNet.

Table 3: Experimental results under various regularization methods. Arrows on the metrics represent the desirable direction. Values represent $\mu \pm \sigma$ obtained from five repetitions, and all values are rounded to two decimal places.

Model & Data	Regularizer	Acc ↑	NLL ↓	ECE ↓	$\| f^{\boldsymbol{W}} \|_2$
VGG-16 & CIFAR-10	Vanilla	92.97 ±0.20	0.35 ±0.01	4.96 ±0.16	9.62 ±0.05
	$\| f^{\boldsymbol{W}} \|_1$	93.07 ±0.08	0.33 ±0.01	4.1 ±0.11	6.62 ±0.01
	$\| f^{\boldsymbol{W}} \|_2^2$	93.06 ±0.03	0.31 ±0.01	4.58 ±0.07	7.44 ±0.03
	$SW_1(\mu^{\boldsymbol{W}}_{\mathcal{D}'}, \nu)$	93.13 ±0.06	0.29 ±0.0	1.9 ±0.07	5.4 ±0.01
	PER	93.1 ±0.17	0.31 ±0.01	4.79 ±0.13	8.49 ±0.03
VGG-16 & CIFAR-100	Vanilla	71.96 ±0.12	1.4 ±0.01	16.9 ±0.09	22.98 ±0.31
	$\| f^{\boldsymbol{W}} \|_1$	72.71 ±0.17	1.44 ±0.01	11.37 ±0.18	9.27 ±0.1
	$\| f^{\boldsymbol{W}} \|_2^2$	72.68 ±0.16	1.31 ±0.01	10.79 ±0.28	9.28 ±0.11
	$SW_1(\mu^{\boldsymbol{W}}_{\mathcal{D}'}, \nu)$	72.26 ±0.23	1.35 ±0.01	12.59 ±0.34	9.72 ±0.02
	PER	72.89 ±0.24	1.34 ±0.01	6.47 ±0.12	7.37 ±0.05