Deep Learning Requires Explicit Regularization for Reliable Predictive Probability

Taejong Joo  
ESTsoft  
Republic of Korea  
tjoo@estsoft.com

Uijung Chung  
ESTsoft  
Republic of Korea  
dict@estsoft.com

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 this traditional role of explicit regularization. Furthermore, explicit regularization does not prevent neural networks from memorizing unnatural patterns, such as random labels, that cannot be generalized. In this work, we revisit the role and importance of explicit regularization methods for generalizing the predictive probability, not just the generalization of the 0-1 loss. Specifically, we present extensive empirical evidence showing the versatility of explicit regularization techniques on improving the reliability of the predictive probability, which enables better uncertainty representation and prevents the overconfidence problem. Our findings present a new direction to improve the predictive probability quality of deterministic neural networks, unlike the mainstream of approaches concentrates on building stochastic representation with Bayesian neural networks, ensemble methods, and hybrid models.

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 [1], and the uncertainty representation ability refers to the ability to represent uncertainty about its predictions. From the deep learning perspective, the reliable predictive probability benefits many downstream tasks such as anomaly detection [2], classification with rejection [3], and exploration in reinforcement learning [4]. More importantly, considering the deep learning system as cognitive automation, the reliable predictive probability plays a significant role in building users’ trust in the automation [5], preventing misuse and disuse of the automation [6], and eventually preventing catastrophic automation failure. Unfortunately, neural networks prune 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, the Bayesian methods express the probability distribution over parameters, in which uncertainty in the parameter space is automatically determined by data. As a result, they can represent uncertainty in prediction by means of providing rich information, such as variance and entropy, about aggregated predictions from different parameter configurations [7, 8]. In this perspective, deterministic neural networks, which select a single parameter configuration so cannot provide the 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, puts significantly more computational overhead compared to the deterministic neural networks. Therefore, the mainstream of improving the predictive probability quality has been an efficient adoption of Bayesian principle into neural networks, so-called Bayesian neural networks, via novel approximation \cite{9,15} and implicitly building the posterior from inherent stochasticity \cite{4,16,17}.

Recent works \cite{3,18,19} discover the hidden gems of label smoothing \cite{20}, mixup \cite{21}, and adversarial training \cite{22} on improving the calibration performance and the uncertainty representation ability. These unexpected findings present a 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 \cite{23,24,25}.

Motivated by these observations, we present a general direction in the regularization perspective to mitigate the unreliable predictive probability problem, rather than devising constructive heuristics or discovering hidden properties of existing methods. Our contributions can be summarized as follows:

- We identify that the unreliable predictive probability is not caused by its deterministic nature, but rather overconfidence predictions on training samples.
- We present a new direction to mitigate the unreliable predictive behavior, which is readily applicable, computationally efficient, and scalable to large-scale models compared to Bayesian neural networks or ensemble methods.
- Our findings give a novel view on the role of regularization for the reliable predictive probability, different from the dominant view on its role in improving generalization performance.

2 Related work

Recent works show that joint modeling of a generative model $p(x)$ along with a classifier $p(y|x)$, or $p(x,y)$ itself, produces the reliable predictive probability \cite{26,28}. Specifically, Alemi et al. \cite{26} argue that modeling stochastic hidden representation through variational information bottleneck principle \cite{29} allows to represent 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 \cite{3,30,31}. 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 focus on building the stochastic representation or the stochastic mapping.

Other works concentrate on structural characteristics of the neural networks. Specifically, Hein et al. \cite{32} 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 multiplier $\lambda$ to an input $x$, for which a neural network produces one dominant entry on $\lambda x$. Verma and Swami \cite{33} point out that the region of the highest predictive uncertainty under the softmax forms a subspace in the logit space, thereby the volume of the area would be negligible. However, our approach suggests that these structural characteristics’ inherent flaws can be easily cured by adding regularization without changing the existing components of neural networks.

From the perspective of the statistical learning theory \cite{34}, a regularization method minimizing some form of complexity measures, e.g., Rademacher complexity \cite{35} or VC-dimension \cite{36}, is a “must” to achieve better generalization of over-parameterized models, which prevents memorization of intricate patterns existing only in training samples. However, the role of capacity control with explicit regularization is challenged by much empirical evidence in deep learning. Specifically, over-parameterized neural networks achieve impressive generalization performance with only implicit regularizations contained in the optimization procedures \cite{37,38} or the structural characteristics \cite{39,41}. Even more, Zhang et al. \cite{42} show that the explicit regularization cannot prevent neural networks from easily fitting random labels that cannot be generalized. 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.
While this standard training procedure results in surprising generalization performance, the resulting uncertainty on out-of-distribution samples, albeit the samples belong to none of the classes seen during training (Figure 1 (b)).

3 Analysis of the unreliable predictive probability in deep learning

3.1 Predictive probability of neural networks

We consider a classification problem with i.i.d. training samples \( D = \{x^{(i)}, y^{(i)}\}_{i=1}^{N} \) drawn from unknown distributions \( P_{x,y} \) whose corresponding random variables are \((x, y)\). We denote \( \mathcal{X} \) as an input space and \( \mathcal{Y} \) as a set of categories \( \{1, 2, \cdots, K\} \). Let \( f^{W} : \mathcal{X} \rightarrow \mathcal{Y} \) be a neural network with parameters \( W \) where \( \mathcal{Z} = \mathbb{R}^{K} \) is a logit space. On top of logit space, the softmax \( \sigma : \mathbb{R}^{K} \rightarrow \Delta^{K-1} \) normalizes exponential of logits, giving interpretation of \( \sigma_{k}(f^{W}(x)) \) as the predictive probability that label of \( x \) belongs to class \( k \) [43]:

\[
\sigma_{k}(f^{W}(x)) = \frac{\exp(z_{k})}{\sum_{i} \exp(z_{i})}, \quad z = f^{W}(x) \tag{1}
\]

where we let \( \sigma_{k}(f^{W}(x)) = \sigma_{k}(f^{W}(x)) \) for brevity.

The de-facto standard for training the neural network is minimizing the cross-entropy loss with stochastic gradient descent (SGD) [44]. For given sample \((x, y)\) and prediction \( \phi^{W}(x) \), the cross-entropy is defined as \( \mathcal{L}(y, \phi^{W}(x)) = -\sum_{k} \mathbb{I}_{y}(k) \log \phi_{k}^{W}(x) \) where \( \mathbb{I}_{A}(\omega) \) is an indicator function taking one if \( \omega \in A \) and zero otherwise. Then, a loss function of \( W \) for the mini-batch samples \( D' \subset D \) is computed by \( \mathcal{L}(W; D') = \mathbb{E}_{(x,y) \sim D'} \left[ -\log \sigma_{y}^{W}(x) \right] \) where \( \mathbb{E}_{D'} \) denotes an empirical mean evaluated on \( D' \). Finally, SGD minimizes the loss by updating parameters via \( W \leftarrow (1 - \lambda) W - \epsilon \nabla_{W} \mathcal{L}(W; D') \) where \( \lambda \) accounts for a weight decay ratio [45] and \( \epsilon \) is a learning rate.

While this standard training procedure results in surprising generalization performance, the resulting neural network often is overconfident and lacks the uncertainty representation ability, which deter deter interpreting the softmax output as the “predictive probability” [4]. Figure 1 illustrates the unreliable predictive behavior of ResNet [46]: the network produces high confidence to misclassified examples (Figure 1 (a)) and provides low predictive entropy on out-of-distribution samples, albeit the samples belong to none of the classes seen during training (Figure 1 (b)).

3.2 Analysis of the unreliable predictive probability

Notably, recalibrating the log-likelihood on unseen samples after training mitigates this problem dramatically [30]. For instance, given a trained network \( f^{W} \) and an extra dataset \( D' \), temperature scaling [47] adjusts temperature \( \tau \) by maximizing the log-likelihood on \( D' \):

\[
\max_{\tau} \sum_{(x,y) \in D'} \log \frac{\exp(f^{W}_{y}(x)/\tau)}{\sum_{j} \exp(f^{W}_{j}(x)/\tau)} \tag{2}
\]

where \( \tau \) controls the smoothness of the softmax output, thereby adjusts the predictive confidence (Figure 1 (c)). This simple procedure makes the softmax output more closely resemble the predictive
We can empirically identify this difference by monitoring the values during training. The empirical probability, for instance, the predictive confidence well-matches its actual accuracy, and the predictive entropy on out-of-distribution samples significantly increases (Figure 1).

Motivated by this observation, we carefully analyze the unreliable predictive behavior of neural networks by anticipating the log-likelihood score on unseen samples. Specifically, we decompose the log-likelihood on random variables $(x, y)$ into two cases whether the predictive class matches the label or not:

$$
E_{x,y}[\log \phi^W_m(x)] = E_{x,y} \left[ \mathbb{I}_{(y)}(m) \log \phi^W_m(x) + \sum_{k \neq m} \mathbb{I}_{(y)}(k) \log \phi^W_k(x) \right] 
$$

$$
\leq E_x \left[ E_{y|x} \left[ \mathbb{I}_{(y)}(m) \right] \log \phi^W_m(x) + (1 - E_{y|x} \left[ \mathbb{I}_{(y)}(m) \right]) \log (1 - \phi^W_m(x)) \right] 
$$

(3)

where $E_{y|x} \left[ \mathbb{I}_{(y)}(m) \right] = p_{y|x}(y = m)$ and $m$ is the predictive class such that $m = \arg \max_k f^W_k(x)$. Note that there exists a multiplier $\alpha \in (0, 1)$ to $(1 - \phi^W_m(x))$ in the second term, which makes the upper bound to the equality, accounting for dispersion of non-maximum probability into $K - 1$ categories. We also note that the log-likelihood is a monotonically increasing function, so we concentrate on the upper bound for the sake of simplicity. Here, given a sample $x \sim x$, the log-likelihood is bounded by the realization of a “stochastic switch” $p_{y|x}(y = m)$ that selects between two “deterministic scores” $\log \phi^W_m(x)$ and $\log(1 - \phi^W_m(x))$.

To scrutinize the score determination mechanism, we note an inherent difference between the deterministic scores and the stochastic switch. The expected deterministic scores, i.e., $E[\log \phi^W_m(x)]$ and $E[\log(1 - \phi^W_m(x))]$, are the property of $f^W$ itself. Therefore, the scores can be anticipated from its estimation $\hat{E}_D[\log \phi^W_m(x)]$ as long as $D_x$ is drawn from $P_x$, in which the difference would be mostly caused by the variance of the Monte-Carlo estimation with a finite sample size. On the other hand, the stochastic switch, i.e., whether the model predicts a label of an unseen sample correctly, depends on the external randomness $y|x$, which makes predicting its behavior on unseen samples significantly challenging. This is because the difference between $\hat{E}_D[\mathbb{I}_{(y)}(m)]$ and $E[\mathbb{I}_{(y)}(m)]$, a.k.a., the generalization gap, is subject to many complex factors (and their interactions), such as input dimensionality, model complexity, and inherent noise (e.g., $[34, 48]$).

We can empirically identify this difference by monitoring the values during training. The empirical means of $L^2$ norm of $f^W$ (Figure 2(a)) and the maximum log-probability $E[\log \phi^W_m(x)]$ (Figure 2(b)), which are the properties of $f^W$, are significantly well preserved those on unseen samples compared to the log-likelihood (Figure 2(b)), which have dependency on the external randomness.

### 3.3 Implication of the analysis

From this perspective, the unreliable predictive probability can be explained by the implicit bias of the cross-entropy minimization. Specifically, the minimum of the cross-entropy is achieved when $f^W_y(x) \to \infty$ and $f^W_k(x) < \infty, \forall k \neq y$. This means that SGD updates $W$ in the direction that

---

We use term “switch” by assuming the noiseless environment, i.e., there is only one true label for each input.
increases $\phi^W_m(x)$ and decreases $\phi^W_k(x), \forall k \neq m$ every time see the example $(x, y)^2$, which in turns make the score $\log \phi^W_m(x)$ near zero and the score $\log(1 - \phi^W_m(x))$ significantly small. For example, Figure 2(c) illustrates the steadily rising trend of misclassified penalty on unseen data caused by increasing confidence on misclassified examples. Therefore, the log-likelihood becomes vulnerable to the notoriously high confidence penalty in the case of misclassified examples.

Therefore, improving the test log-likelihood requires reducing the impact of the confidence penalty, which can be achieved by reducing confidence on misclassified examples or decreasing misclassification rate. In this work, we focus on reducing the predictive confidence on training samples, thereby the unseen samples, through explicit regularization techniques and show its effectiveness for the rest of the paper. This is because empirical evidence (cf. [47]) shows that the improved generalization performance frequently worsens the predictive probability quality, which may be caused by that an increased capacity enables to fit training samples more confidently. For example, Figure 3 shows that increasing model capacity reduces the misclassification rate but worsens the calibration performance, called expected calibration error (ECE; cf. metrics in Section 4).

4 Experiment

Setup. Our main experimental model is the (pre-activation) ResNet [46] trained on CIFAR [49], which is one of the most prevalent basis models in many state-of-the-art architectures [50, 51]. We also present the VGG [52] as a representative of models without residual connection in appendix B, in which we observe similar results to ResNet. We performed all experiments with a single GPU and trained our model with the standard training procedure based on [46] except learning rate warm-up for the first five epochs, clipping gradient when its norm exceeds one, and extra validation set of 10,000 samples split from the training set. We describe a detailed setup in appendix A.

Metrics. To precisely evaluate the reliability of the predictive probability, we employ various metrics commonly used in literature [3, 30, 47]:

- Negative log-likelihood (NLL) evaluates how well the predictive probability explains the test data $D^T$, which is computed by: $-\hat{E}_{D^T} \log \phi^W_y(x)$. NLL has a desirable property that its optimal score is achieved and only if $\phi^W(x)$ perfectly match $p(y|x)$.
- Expected calibration error (ECE) [53] evaluates how well the predictive confidence matches its actual accuracy. Specifically, ECE on $D^T$ is computed by binning predictions into $M$-groups based on their confidences such that $G_i = \{x : i/M < \max_k \phi^W_k(x) \leq (1 + i)/M, x \in D^T\}$, then averaging their calibration scores by $\sum_{i=1}^M |G_i| [\bar{\text{acc}}(G_i) - \bar{\text{conf}}(G_i)]$ where $\bar{\text{acc}}(G_i)$ and $\bar{\text{conf}}(G_i)$ are average accuracy and confidence of predictions in group $G_i$, respectively.
- Predictive entropy on out-of-distribution samples evaluates how well the predictive uncertainty represents their ignorance, which is computed by $H[\phi^W(x)]$ for the out-of-distribution samples. Here, the reliable predictions ought to produce the highest entropy as the samples belong to none-of-classes seen during training.

4.1 Confidence control by weight decay

The simplest way to constrain the confidence may conceivably be controlling the strength of weight decay [45] that can encourage producing less extreme outputs by shrinking weights. Therefore, we first explore the confidence control by varying the weight decay ratio $\lambda$, conjecturing that the weight decay ratio, e.g., $\lambda = 0.0001$ in ResNet, is too small to prevent overconfident predictions. Figure 4 (upper) illustrates the impact of $\lambda$ on changes in generalization performance and calibration.

---

2This holds when it corrects the answer, for which modern neural networks easily achieve for most of samples in the early stage of training.

![Figure 3: Accuracy-ECE comparison of different models on ImageNet. Points connected to the same line represent the same model family with different capacity.](image-url)
performance. We can see that ECE decreases to some extent, as the ratio increases. However, the ECE improvement becomes inversely proportional to a generalization performance improvement when the decaying 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 training behavior under different weight decay rates and comparing their impacts to temperature scaling. In section [5.2], we have shown that the temperature scaling mitigates the impact of the high confidence penalty on the log-likelihood, thereby improving the reliability of the predictive distribution. The temperature scaling achieves this by dividing logits with a scalar, which means that it controls the $L^2$ norm of the function; that is, $\| f^W/\tau \|_2 = \| f^W \|_2/\tau$. For this reason, we monitor the evolution of $L^2$ norm, and compare the value with $\| f^W/\tau^* \|_2$ where $W$ is the weight of the neural network with $\lambda = 0.0001$ and $\tau^* = \arg \max_{\tau} E_P \left[ \log(\phi^W(x)/\tau) \right]$ that is obtained by “leaking” the test set $D^T$.

Figure 4 shows that the SGD with various decay ratios $\lambda$ finds only the trivial solution or an infeasible solution in the perspective of the following optimization problem:

$$\min_W E_P \left[ l_{CE}(y, \phi^W(x)) \right] \ \text{s.t.} \ \| f^W \|_2 \leq \| f^W/\tau^* \|_2$$

(4)

which indicates that confidence control by adjusting the weight decay ratio is the significantly challenging optimization problem. Specifically, Figure 4 (lower) shows that the $L^2$ norm becomes zero when the decay ratio $\lambda \geq 0.003$, which means that all weights collapse to zero, i.e., the trivial solution. This happens when the decay ratio overwhelms the gradient of the cross-entropy, e.g., around at the epoch 50 under $\lambda = 0.003$ (Figure 4 (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 higher than $\| f^W/\tau^* \|_2$, which correspond to infeasible solutions (Figure 4 (lower)). These results may seem natural because the weight decay does not consider constraints about the predictive confidence. Therefore, we explore a way to add a regularization loss that explicitly concerns the predictive confidence, e.g., the constraint in equation 4.

4.2 Direct confidence control by explicit regularization

In this subsection, we examine two types of regularizers that directly constrain the predictive confidence on the input probability distribution space $P(\mathcal{X})$, whose effectiveness on improving the reliability of the predictive distribution has not explored yet.

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

$$\| f^W \|_p = \left( \int_\mathcal{X} |f^W(x)|^p dP_x(x) \right)^{1/p} < \infty$$

(5)

Here, we note that the norm is computed with respect to the input generating distribution $P_x$, which allows to control how the function $f^W$ actually behaves on. Since $P_x$ is unknown, so it is approximated by the Monte-Carlo approximation with mini-batch samples. Then, the approximate function norm can be computed by $\| f^W \|_p \approx \frac{1}{m} \sum_i |f^W(x^{(i)})|^p$. By penalizing the complexity in terms of the $L^p$ norm, continuous increase in the leading entry of logit towards infinity, or continuous decreases in the non-leading entry to negative infinity, can be prevented (cf. Section 3.3). In this paper, we examine $\| f^W \|_1$ and $\| f^W \|_2^2$ regularization losses.

Regularization in the probability distribution space. The second approach regards $f^W(x)$ as a random variable, then minimize its distance to a simple distribution, i.e., standard normal distribution.\footnote{We note the possibility of more theoretically ground confidence control by encoding more meaningful information into the target distribution, e.g., determination of the precision parameter, leaving it as future work.}
In this work, 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 for the empirical distribution. We refer Peyré et al. [54] for more detailed explanations about this metric. Specifically, given mini-batch samples \( D' = \{ x^{(i)} \}^m_{i=1} \), let an empirical measure of logits be \( \mu_{D'}(\lambda) = \frac{1}{m} \sum_i \mathbb{1}_\lambda (f^W(x^{(i)})) \) and the standard Gaussian measure on \( Z \) be \( \nu(\lambda) = \frac{1}{2\pi} \int_{\mathbb{R}^d} \exp \left(-\frac{1}{2} \| z \|^2 \right) dz \). Then, the sliced Wasserstein distance can be computed by:

\[
SW_1(\mu_{D'}, \nu) = \int_{S^{K-1}} \int_{-\infty}^{\infty} \left| F_{\mu_\theta}(x) - \frac{1}{m} \sum_{i=1}^m \mathbb{1}_{(\infty, x)}(z^{(i)}(\theta)) \right| dx d\lambda(\theta)
\]  

(6)

where \( z^{(i)} = f^W(x^{(i)}) \), \( \lambda \) is a uniform measure on the unit sphere \( S^{K-1} \), and \( \mu_\theta \) is a measure obtained by projecting \( \mu_{D'} \) at angle \( \theta \). Therefore, the confident predictions, each of which involves one dominant entry, result in a significant penalty as the empirical distribution consists of such predictions is far from the standard normal distribution. Projected error function regularization (PER) [55] further simplify the \( SW_1(\mu_{D'}, \nu) \) by applying Minkowski inequality to the above equation. As a result, the gradient of PER resembles the gradient of Huber loss [56] in the projected space, which allows the robust norm measurement 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 [55].

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 to 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.

We split 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 uncertainty representation ability on out-of-distribution samples. Since out-of-distribution samples don’t belong to any categories, the neural network should produce the answer of “I don’t know.” Figure 5 illustrates predictive uncertainty of ResNet-50 with respect to CIFAR-100 (in-distribution) and SVHN [57] (out-of-distribution). 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, ResNet under explicit regularization successfully gathers a mass of predictive uncertainty for SVHN samples on the region around maximum-entropy ($\log 100 \approx 4.6$).

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 [4], because other methods based variational inference [10–12] or MCMC [14, 15] 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 [3] with 5 number of ensembles, i.e., 5x more training and inference time. Figure 5 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.

5 Conclusion

In this works, we show that “deep learning requires explicit regularization for reliable predictive probability.” Specifically, we systematically analyze contributing factors for the unreliable predictive probability by decomposing the log-likelihood into the stochastic switch, i.e., whether the predictive class matches the label, that chooses 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 improving calibration, uncertainty representation, and even the test accuracy.

Our findings present a novel view on the role and importance of explicit regularization for improving the reliability of the predictive probability of neural networks. 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 metrics based on stochastic representation on the predictive probability space, such as mutual information measuring epistemic uncertainty [58], 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., [25, 29, 60].
Broader Impact

This work shows the effectiveness of explicit regularization methods on improving the reliable predictive probability of deep neural networks, which helps the neural networks to produce more calibrated predictions and represent predictive uncertainty better. As the regularization-based methods are more computationally efficient and readily applicable compared to those of the Bayesian or ensemble methods, our findings would encourage many practitioners to accommodate the reliable deep learning models. Once we view the deep learning systems as cognitive automation, meaning that it aids human decision-making processes or replace a part of cognitive tasks previously done by humans; this would result in the better predictive probability, which means the better feedback of explaining what’s going on, the situations when the automation becomes uncertain, and unexpected anomalies. This form of appropriate feedback can prevent the misuse and disuse of automation, including automation failures or even catastrophic accidents in safety-critical domains [5][6][61].

Conversely, the wide adoption of the reliable predictive probability models could put extra training burdens on humans because interpreting information from predictive uncertainty or calibrated prediction requires human operators to be well-trained to leverage the benefits of such information [62]. Besides, providing the uncertainty information or confidence level may increase humans’ cognitive workload, which can result in attention distraction and task performance degradation [63]. Finally, our findings may inherit biases contained in the standard classification benchmark environment, for which we follow for precise evaluation.

References

[1] A Philip Dawid. The well-calibrated Bayesian. Journal of the American Statistical Association, 77(379):605–610, 1982.
[2] 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.
[3] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In Advances in Neural Information Processing Systems, 2017.
[4] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In International Conference on Machine Learning, 2016.
[5] John Lee and Neville Moray. Trust, control strategies and allocation of function in human-machine systems. Ergonomics, 35(10):1243–1270, 1992.
[6] Raja Parasuraman and Victor Riley. Humans and automation: Use, misuse, disuse, abuse. Human Factors, 39(2):230–253, 1997.
[7] David JC MacKay. A practical Bayesian framework for backpropagation networks. Neural Computation, 4(3):448–472, 1992.
[8] Radford M Neal. Bayesian learning via stochastic dynamics. In Advances in Neural Information Processing Systems, 1993.
[9] Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In Annual Conference on Computational Learning Theory, 1993.

5 Here, we assume the involvement of deep learning in the real-world situation as a hybrid system, wherein humans and automation cooperate to solve some forms of cognitive tasks. This assumption is valid in a sense that the fully autonomous system, which requires an almost perfect level of success rates and a higher level of cognitive tasks such as reasoning, adapt to continuously changing environment, and dealing with a complete anomaly, would not be possible at least with the current level of deep learning. Indeed, the cooperative cognitive tasks already prevalent around the world. For example, the autonomous driving software handles the normal driving situation while a human driver supervises its behavior and take the driving authority under uncertainty or exceptional circumstances. Even in simple image labeling tasks, humans make the final decision to complement the imperfect test accuracy of neural networks.
[10] Alex Graves. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems*, 2011.

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

[12] 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.

[13] José Miguel Hernández-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *International Conference on Machine Learning*, 2015.

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

[15] Ruqi Zhang, Chunyuan Li, Jiaoyun Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclic stochastic gradient MCMC for Bayesian deep learning. In *International Conference on Learning Representations*, 2020.

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

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

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

[19] 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.

[20] 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.

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

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

[23] 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.

[24] Kazuki Osawa, Siddharth Swaroop, Mohammad Emiiyaz 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.

[25] Taejong Joo, Uijung Chung, and Min-Gwan Seo. Being Bayesian about categorical probability. *arXiv preprint arXiv:2002.07965*, 2020.

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

[27] 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.

[28] 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.
[29] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In International Conference on Learning Representations, 2017.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] Vladimir Vapnik. The Nature of Statistical Learning Theory. Springer-Verlag, 1995.

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

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

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

[38] 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.

[39] 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.

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

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

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

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

[44] Herbert Robbins and Sutton Monro. A stochastic approximation method. The Annals of Mathematical Statistics, pages 400–407, 1951.

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

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

[47] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In International Conference on Machine Learning, 2017.
[48] 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.

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

[50] 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.

[51] 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.

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

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

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

[55] 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*, 2020.

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

[57] 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.

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

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

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

[61] Donald A Norman. The ‘problem’ with automation: Inappropriate feedback and interaction, not ‘over-automation’. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, 327(1241):585–593, 1990.

[62] Susan S Kirschenbaum, J Gregory Trafton, Christian D Schunn, and Susan B Trickett. Visualizing uncertainty: The impact on performance. *Human Factors*, 56(3):509–520, 2014.

[63] Alexander Kunze, Stephen J Summerskill, Russell Marshall, and Ashleigh J Filtness. Automation transparency: Implications of uncertainty communication for human-automation interaction and interfaces. *Ergonomics*, 62(3):345–360, 2019.

[64] 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.
## 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 [64]; 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 [52].

**Hyperparameters.** We searched four regularization loss coefficients for each method, and chose best one based on validation set accuracy (Table A1). 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 [55].

| Regularizer | VGG-16 & CIFAR-10 | VGG-16 & CIFAR-100 | ResNet-50 & CIFAR-10 | ResNet-50 & CIFAR-100 |
|-------------|------------------|-------------------|---------------------|---------------------|
| $\| f^W \|_1$ | 0.01             | 0.03              | 0.01                | 0.01                |
| $\| f^W \|_2$ | 0.003            | 0.01              | 0.003               | 0.01                |
| SW$_1(\mu_D', \nu)$ | 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 A2), 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 $\| f^W \|_2$ of VGG are reduced by almost 50% compared to those of ResNet.

| Model & Data | Regularizer | Acc ↑ | NLL ↓ | ECE ↓ | $\| f^W \|_2$ |
|--------------|-------------|-------|-------|-------|--------------|
| VGG-16 & CIFAR-10 | Vanilla     | 92.97 ± 0.20 | 0.35 ± 0.01 | 4.96 ± 0.16 | 9.62 ± 0.05 |
|              | $\| f^W \|_1$ | 93.07 ± 0.08 | 0.33 ± 0.01 | 4.1 ± 0.11  | 6.62 ± 0.01  |
|              | $\| f^W \|_2$ | 93.06 ± 0.03 | 0.31 ± 0.01 | 4.58 ± 0.07 | 7.44 ± 0.03  |
|              | SW$_1(\mu_D', \nu)$ | 93.13 ± 0.06 | 0.29 ± 0.00 | 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 |
|              | $\| f^W \|_1$ | 72.71 ± 0.17 | 1.44 ± 0.01 | 11.37 ± 0.18 | 9.27 ± 0.1   |
|              | $\| f^W \|_2$ | 72.68 ± 0.16 | 1.31 ± 0.01 | 10.79 ± 0.28 | 9.28 ± 0.11  |
|              | SW$_1(\mu_D', \nu)$ | 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  |