Energy-Based Open-World Uncertainty Modeling
for Confidence Calibration

Yezhen Wang[1*]   Bo Li[1*]   Tong Che[2*]   Kaiyang Zhou[3]
Ziwei Liu[3]   Dongsheng Li[1]

[1]Microsoft Research Asia   [2]MILA   [3]S-Lab, Nanyang Technological University

Abstract

*Confidence calibration is of great importance to the reliability of decisions made by machine learning systems. However, discriminative classifiers based on deep neural networks are often criticized for producing overconfident predictions that fail to reflect the true correctness likelihood of classification accuracy. We argue that such an inability to model uncertainty is mainly caused by the closed-world nature in softmax: a model trained by the cross-entropy loss will be forced to classify input into one of K pre-defined categories with high probability. To address this problem, we for the first time propose a novel $K+1$-way softmax formulation, which incorporates the modeling of open-world uncertainty as the extra dimension. To unify the learning of the original K-way classification task and the extra dimension that models uncertainty, we **1)** propose a novel energy-based objective function, and moreover, **2)** theoretically prove that optimizing such an objective essentially forces the extra dimension to capture the marginal data distribution. Extensive experiments show that our approach, Energy-based Open-World Softmax (EOW-Softmax), is superior to existing state-of-the-art methods in improving confidence calibration.*

1. Introduction

Given the considerable success achieved so far by deep neural networks (DNNs), one might be wondering if DNN-based systems can be readily deployed to solve real-world problems. On the one hand, DNNs can achieve high accuracy if trained with large-scale datasets [15]. But on the other hand, contemporary DNNs are often criticized for producing overconfident predictions [14], which fail to represent the true correctness likelihood of accuracy. This has raised concerns over safety and reliability for

using machine learning systems in real-world scenarios. Having a confidence-calibrated system is critical. For instance, in healthcare applications, the intelligence system should produce low-confidence predictions when it is uncertain about the input—say they differ significantly from the training data—so the decision-making process can be transferred to human doctors for more accurate diagnosis and safer handling. Research on confidence calibration for DNNs has received increasing attention in recent years [14, 27, 19, 23, 27]. Since most classifiers are based on softmax, a common practice to improve calibration is to insert a temperature scaling parameter to the softmax function and adjust it in a validation set [14]. Besides, methods like Smoothing labels [36, 28], which essentially combines the one-hot ground-truth vector with a uniform distribution, has also been shown effective in improving calibration.

However, most existing confidence calibration methods have overlooked the underlying problem that causes neural network classifiers to generate overconfident predictions, i.e. the inability to model *uncertainty* in output probabilities. We argue that the culprit for causing such a problem is the closed-world nature in softmax [2, 34]. This is easy to understand: during training the model is asked to classify input into one of K pre-defined categories with high probability (due to the cross-entropy loss), and as such, the model has no choice but to assign one of the K categories to any unseen data, likely with high probability as well.

A potential countermeasure is to adopt a $K + 1$-way formulation where the new category can represent uncertainty about the input data. In this way, the K classification scores might be better regularized, and hence better calibrated. However, learning such a classifier is challenging as we do not have access to those data with the $K + 1$-th label, thus lacking supervision to teach the network when to give low/high confidence. Furthermore, designing the extra dimension is a non-trivial task as it is directly linked to the formulation of the learning objective. It is also unclear how such a dimension should be constructed, e.g., to design it as another logit produced by the same network or an indepen-

---
*Equal contribution. Ordered by dice rolling. Correspondence to yezhen.wang0305@gmail.com.

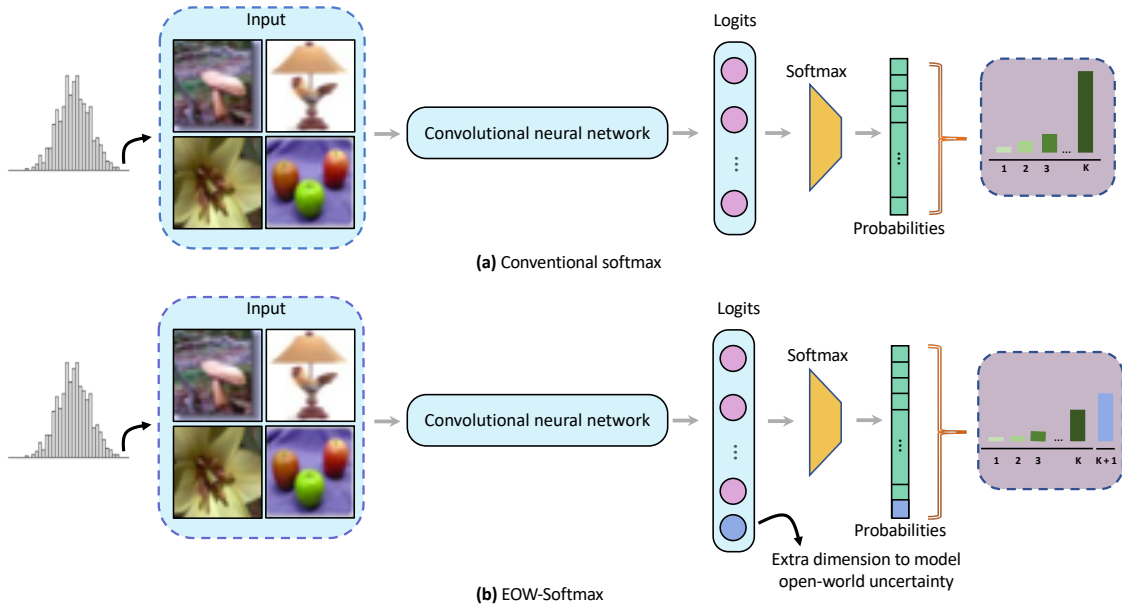**(a)** Conventional softmax

(b) EOW-Softmax

Figure 1. Comparison between (a) the conventional softmax and (b) our proposed Energy-based Open-World softmax (EOW-Softmax). Our new formulation introduces an extra dimension to model uncertainty, which is supposed to produce high scores when the input deviates from the training data distribution. In this way, the original K classification scores can be well calibrated.

dent branch that regresses to uncertainty.

In this paper, we propose *Energy-based Open-World Softmax* (EOW-Softmax), a novel approach that introduces a $K + 1$-way softmax based on energy functions [25]. Specifically, the neural network classifier is designed to produce $K + 1$ logits, where the first K dimensions encode the scores for the original K-way classification task, while the extra dimension aims to model open-world uncertainty. See Figure 1 for a comparison between a model based on the conventional softmax and that based on our EOW-Softmax. Besides, we resort to an energy-based $K + 1$-way classification objective function to unify the learning of the K-way classification task and the uncertainty modeling. Further more, we theoretically justify that optimizing the proposed objective function essentially forces the summation of original K softmax scores ($K + 1$ scores in total) to be directly proportional to the marginal density $p(x)$, hence explaining why our EOW-Softmax helps calibrate a model's confidence estimates.

The **contributions** of this paper are summarized as follows. **1)** First, we overcome the closed-world softmax problem by transforming the conventional K-way softmax to a novel $K + 1$-way formulation, where the extra dimension is designed to model open-world uncertainty. **2)** Second, a novel energy-based objective function is developed to unify the learning of the original K-way classification task and the uncertainty modeling. **3)** A theoretical proof is further provided to explain why our learning objective can help the network capture uncertainty. **4)** Finally, we conduct extensive experiments on standard benchmark datasets to demon-

strate that our method can lead to a better calibrated model compared with other state-of-the-arts.

2. Related Works

Confidence Calibration   With the emergence of deep learning technologies and their wide successes, concerns over whether they are reliable to be deployed in practice have also arisen. This is because researchers have found that contemporary deep neural networks (DNNs) often produce overconfident predictions [14], even on input images that are totally unrecognizable to humans [32]. Many approaches for improving confidence calibration have been developed. A widely used method is temperature scaling [14, 27, 19], which inserts a scaling parameter to the softmax formulation (called 'temperature') and adjusts it in a validation set with a goal to 'soften' the softmax probabilities. Regularization methods, such as label smoothing [36] and Mixup [37], have also been demonstrated effective in improving calibration. In particular, label smoothing modifies the ground-truth labels by fusing them with a uniform distribution, essentially forcing neural networks to produce 'more flattened' probabilities; whereas Mixup is a data augmentation method that randomly mixes two instances at both the image and label space, with a byproduct effect of improving calibration. Bayesian methods have also been explored for calibration. For instance, Monte Carlo Dropout [7] applies dropout in both training and testing; Deep Ensembles [23] uses as prediction the output averaged over an ensemble of models. Adding adversarial perturba-

tions to the input has been found effective in smoothing the output probabilities [23, 27]. In [26], a GAN model [11] is trained to generate out-of-distribution (OOD) data and the classifier is encouraged to produce low-confidence probabilities on these data. Such an idea has also been investigated in [16] where adversarial perturbations are utilized to synthesize OOD data. In [12], a Joint Energy-based Model (JEM) is proposed to improve calibration by learning the joint distribution based on energy functions [25]. A recent work [38] suggests that calibrating confidence across multiple domains is beneficial to OOD generalization [42]. Studies on why neural networks produce overconfident predictions have also been covered in the literature. In [16], the authors suggest that ReLU neural networks are essentially piecewise linear functions, thus explaining why OOD data can easily cause softmax classifiers to generate highly confident output. In [30], the authors identify that data variance and model curvature cause most generative models to assign high density to OOD data. The authors in [34] point out that the overconfidence issue is related to the closed-world assumption in softmax, and design a distance-based one-vs-all (OvA) classifier as the countermeasure.

Two works related to ours are JEM [12] and the OvA classifier [34]. Compared with JEM, our approach is much easier to train because we only need to optimize a *single* classification objective to achieve both discriminative classifier learning and generative modeling (see Sec. 3.3), while JEM has to simultaneously optimize two separate objectives. Moreover, JEM has ignored the closed-world softmax issue, which is addressed in this work with an augmented softmax. Compared with the OvA classifier, our approach is significantly different: we endow the classifier with the ability to model open-world uncertainty, which is attributed to the extra dimension in softmax learned via a novel energy-based objective function to capture the marginal data distribution; in contrast, the OvA classifier converts the K-way classification problem into multiple binary classification problems.

Energy-Based Models (EBMs) have been widely used in the area of generative modeling [41, 8, 1, 4]. The basic idea in EBMs is to learn dependencies between variables (e.g., images and labels) represented using energy functions; and to assign low energies to correct configurations while give high energies to incorrect ones [25]. However, training EBMs, especially on high-dimensional data like images, has been notoriously hard due to sampling issues [21, 13]. A widely used sampler is Stochastic Gradient Langevin Dynamics (SGLD) [39], which injects noises to the parameter update and anneals the stepsize during the course of training. Following prior work [33, 6, 12], we also leverage SGLD to optimize our energy-based objective function.

3. Methodology

According to [34], we argue that the culprit for causing the overconfidence problem in most neural network classifiers' output is the closed-world nature in softmax. As a result, a model trained by the cross-entropy loss has to pick one of K pre-defined categories with high confidence. To overcome this problem, we propose *Energy-based Open-World Softmax*, or EOW-Softmax, to regularize the K-way classification scores in such a way that allows their confidence to be calibrated. The main idea in EOW-Softmax is to re-formulate the original K-way classification task as a novel K+1-way classification problem, where the extra dimension is designed to model *open-world uncertainty*. To learn such a K+1-way classifier in an end-to-end manner, we propose a novel energy-based objective function, which essentially forces the extra dimension to be negatively correlated to the marginal data distribution. In this way, the K classification scores are automatically calibrated to be less confident over input fallen beyond the training data distribution. See Figure 2 for an overview of our model architecture. The rest of this section are organized as follows. In Sec. 3.1, we provide a brief background on energy-based models (EBMs), which are required by our approach to construct the objective function. In Sec. 3.2, we discuss in detail the design of EOW-Softmax. Sec. 3.3 gives a theoretical insight on why EOW-Softmax can help calibrate a model's confidence estimates.

3.1. A Brief Background on EBMs

The main building block in EBMs is an energy function $E_\theta : \mathbb{R}^D \to \mathbb{R}$ (parameterized by θ), which aims to map a D-dimensional datapoint to a scalar.[1] The learning is designed in such a way that E_θ can assign low energies to observed configurations of variables while give high energies to unobserved ones [25]. With E_θ, any probability density $p(x)$ for $x \in \mathbb{R}^D$ in an EBM can be written as

$$p_\theta(x) = \frac{\exp(-E_\theta(x))}{Z(\theta)}, \qquad (1)$$

where $Z(\theta) = \int_x \exp(-E_\theta(x))$ denotes the normalizing constant, also known as the partition function. An EBM can be represented by using any function as long as the function can generate a single scalar given some input x. In this work, we assume E_θ is represented by a deep neural network. People usually adopt gradient estimation to optimize an EBMs [21, 13] and sample data from it by Markov Chain Monte Carlo (MCMC) methods [9, 10, 18, 39].

---

[1]When the input is an image, D can be understood as the length of the flattened tensor.
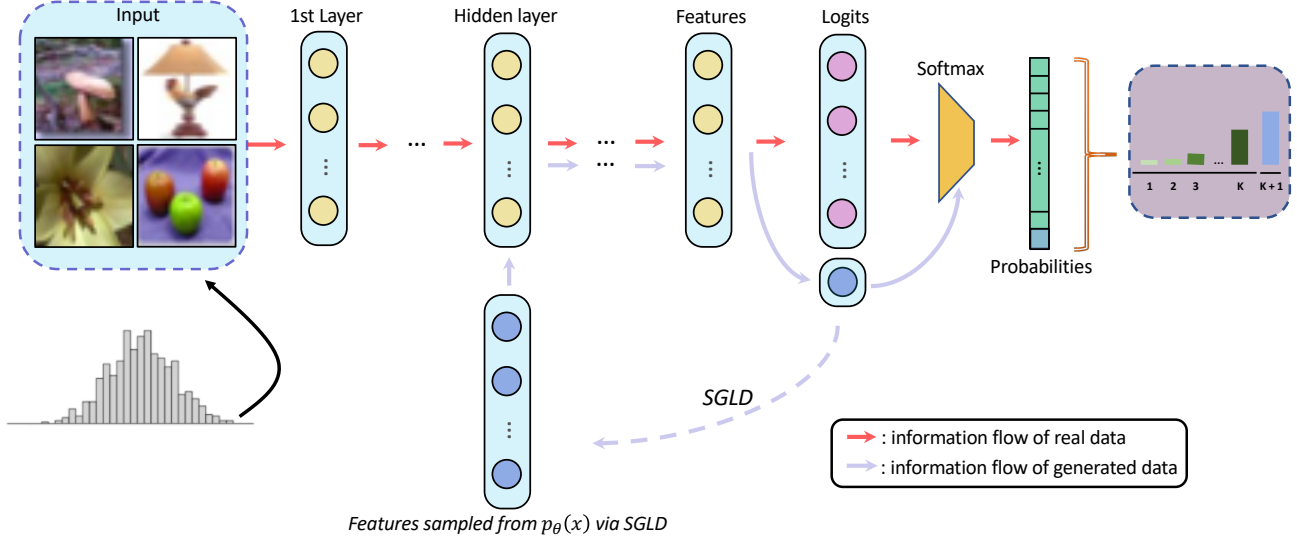
Figure 2. Model architecture of our approach *EOW-Softmax*. The extra dimension introduced in the augmented softmax (dashed) is learned using an energy-based function to model open-world uncertainty, such that it can assign high uncertainty scores to abnormal input far away from the training data distribution, which in turn lower the classifier's confidence on the original K-way classification task. Note that the sampling in SGLD is performed on the latent (feature) space rather than the input (image) space.

3.2. Energy-Based Open-World Softmax

Open-World Softmax   As discussed before, conventional softmax-based classifiers lack the ability to model open-world uncertainty. To address this problem, we design a neural network classifier to output probabilities on $K + 1$ categories (see Figure 2), with the $K+1$-th score representing open-world uncertainty—the network should be able to produce high uncertainty scores to abnormal input, which in turn can lower the confidence on the original K categories' prediction. Let $f_\theta : \mathbb{R}^D \to \mathbb{R}^{K+1}$ be our neural network model (excluding the softmax layer), which produces $K + 1$ logits, and $f_\theta(x)[i]$ the i-th logit given input x, with $i \in \{1, ..., K, K + 1\}$. The output probabilities can then be obtained by passing these $K + 1$ logits to a softmax normalization layer, i.e.

$$h_\theta(x)[i] = \frac{\exp(f_\theta(x)[i])}{\sum_{j=1}^{K+1} \exp(f_\theta(x)[j])}, \qquad (2)$$

where h_θ is the combination of the neural network f_θ and the softmax normalization layer.

Energy-Based Learning Objective   Now the question is how to design a learning objective that allows $h_\theta(x)[K+1]$ to encode uncertainty? Our idea here is to associate the score of $h_\theta(x)[K + 1]$ to the marginal data distribution. Intuitively, when the input comes from within the training data distribution $p(x)$, the model is supposed to be confident in its decision, and therefore, $h_\theta(x)[K + 1]$ should be low (conversely, $\sum_{i=1}^{K} h_\theta(x)[i]$ should be high). If the input deviates from the training data distribution, the model should

become uncertain about whether its decision is correct. In this case, $h_\theta(x)[K + 1]$ should be high to indicate a higher level of uncertainty, which naturally forces $\sum_{i=1}^{K} h_\theta(x)[i]$ to stay low (due to the softmax normalization). However, directly training $h_\theta(x)[K + 1]$ to capture the marginal distribution $p(x)$ (i.e. generative modeling) is difficult [30]. Instead, we propose a novel learning objective with the help of EBMs [25]. First, we define our energy function as

$$E_\theta(x) = \log h_\theta(x)[K + 1]. \qquad (3)$$

Then, our energy-based objective function is defined as

$$\min_\theta \mathbb{E}_{p(x)}\left[-\log h_\theta(x)[y]\right] + \lambda \mathbb{E}_{p_{\bar\theta}(x)}\left[-\log h_\theta(x)[K+1]\right], \qquad (4)$$

where $\lambda > 0$ is a hyper-parameter; the first term is the maximum log-likelihood objective for the K-way classification task using the ground-truth label y; the second term can also be seen as maximum log-likelihood objective—for recognizing data sampled from $p_{\bar\theta}(x)$. Note that $p_{\bar\theta}(x)$ denotes the model distribution with frozen parameters of $p_\theta(x)$ of current iteration. We will show later in Sec. 3.3 that optimizing Eq. (4) can actually lead the summation of rest K softmax scores of original classes to be directly proportional to the marginal density $p(x)$, which in turn can make the $K + 1$-th softmax score be negatively correlated to $p(x)$.

SGLD-Based Optimization   We approximate the expectation in the second term in Eq. (4) using a sampler based on Stochastic Gradient Langevin Dynamics (SGLD) [39].

Specifically, the SGLD sampling process follows

$$z_{t+1} = z_t - \frac{\alpha}{2}\frac{\partial E_\theta(z_t)}{\partial z_t} + \sqrt{\alpha}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \quad (5)$$

where t denotes the SGLD iteration, α the step-size, and ϵ a random noise sampled from a normal distribution. In practice, α is usually fixed as a constant.

Most SGLD-based methods draw samples from the image space. This forces the information to flow through the entire neural network, which is computationally expensive. Inspired by [4], we choose to draw samples from the latent space (see Figure 2). Therefore, z in Eq. (5) represents features rather than an image. Such a design significantly accelerates the training since the information only goes partially through the network model, which allows much deeper architectures such as ResNet-101 [15] to fit into limited resources. Moreover, the latent space is typically smoother than the image space [3], which facilitates the estimate of gradients.

3.3. Theoretical Insight

In order to prove that our objective can force $h_\theta(x)[K+1]$ to be negatively correlated with $p(x)$, i.e. representing uncertainty, we tend to show that in theory, optimizing our objective in Eq. (4) is equivalent to minimize the KL divergence between $p(x)$ and another EBM-modeled distribution $q_\theta(x)^2$, where $q_\theta(x)$ is defined by energy function

$$E'_\theta(x) = -\log \sum_{i=1}^{K} h_\theta(x)[i]. \quad (6)$$

To this end, we introduce an extra objective

$$\min_\theta \mathbb{E}_{p(x)}\left[-\log h_\theta(x)[y]\right] + \mathbb{E}_{q_{\bar\theta}(x)}\left[\log \sum_{i=1}^{K} h_\theta(x)[i]\right], \quad (7)$$

Similar to Eq. (4), here $\bar\theta$ means the parameters are frozen. We will show that optimizing Eq. (4) essentially optimize Eq. (7) and optimizing Eq. (7) is an equivalent of $\min D_{KL}(p(x)||q_\theta(x))$ in following Proposition 1 and Theorem 1, respectively.

Proposition 1. *Given two EBMs $p_{\bar\theta}(x)$ and $q_{\bar\theta}(x)$ with energy functions defined in Eqs. (3) and (6) respectively, the optimization of Eq. (4) is actually equivalent to optimize a combination of one K-way classification objective and Eq. (7) with some suitable coefficient μ.*

Proof. Since the first optimization term in Eq. (7) is identical with Eq. (4) as well as the objective of maximum log-likelihood of K-way classification problems, we only need

to consider the second terms in both equations and prove that they are equivalent to each other. Specifically, for the gradient of the second term of Eq. (7), we have[3]

$$\frac{\partial}{\partial\theta}\mathbb{E}_{q_{\bar\theta}(x)}\left[\log\sum_{i=1}^{K} h_\theta(x)[i]\right] \quad (8)$$

$$= \int_x q_{\bar\theta}(x)\frac{\partial}{\partial\theta}\log\sum_{i=1}^{K} h_\theta(x)[i] \quad (9)$$

$$= \int_x \frac{\sum_{i=1}^{K} h_\theta(x)[i]}{Z'(\theta)} \cdot \frac{\frac{\partial}{\partial\theta}\sum_{i=1}^{K} h_\theta(x)[i]}{\sum_{i=1}^{K} h_\theta(x)[i]} \quad (10)$$

$$= \frac{1}{Z'(\theta)}\int_x \frac{\partial}{\partial\theta}\sum_{i=1}^{K} h_\theta(x)[i] \quad (11)$$

$$= -\frac{1}{Z'(\theta)}\int_x \frac{\partial}{\partial\theta}h_\theta(x)[K+1] \quad (12)$$

$$= -\frac{Z(\theta)}{Z'(\theta)}\int_x \frac{h_\theta(x)[K+1]}{Z(\theta)} \cdot \frac{\frac{\partial}{\partial\theta}h_\theta(x)[K+1]}{h_\theta(x)[K+1]} \quad (13)$$

$$= -\frac{Z(\theta)}{Z'(\theta)}\int_x p_{\bar\theta}(x)\frac{\partial}{\partial\theta}\log h_\theta(x)[K+1], \quad (14)$$

where $Z'(\theta)$ and $Z(\theta)$ represents the partition functions of $q_\theta(x)$ and $p_\theta(x)$ respectively. If we use μ to denote $\frac{Z(\theta)}{Z'(\theta)}$, we can restate the Eq. (14) as

$$\mu\frac{\partial}{\partial\theta}\mathbb{E}_{p_{\bar\theta}(x)}\left[-\log h_\theta(x)[K+1]\right],$$

which is exactly the gradient of Eq. (4)'s second term. As a result, the objective of Eq. (4) and of Eq. (7) are same under a suitable coefficient μ in each iteration. Further more, remembering that the first term of Eq. (4) is an equivalent of the objective of maximum log-likelihood of K-way classification problems, we consequently conclude that optimizing our objective of Eq. (4) essentially optimize Eq. (7) and a K-way classification objective. $\square$

According to Proposition 1, we know that our learning objective of Eq. (4) can optimize the discriminative K-way classification objective and the generative modeling objective in a unified discriminative objective, which has never been explored in existing confidence calibration work.

Moreover, if we can further prove that optimizing Eq. (7) is *de facto* an equivalent of minimizing the KL-divergence between $p(x)$ and the distribution $q_\theta(x)$, then our objective

---

[2]It is worth noting that this $q_\theta(x)$ is not the modeled EBMs $p_\theta(x)$ in Sec. 3.2.

[3]The equality between Eqs. (9) and (10) holds because $q_{\bar\theta}(x) = \frac{\sum_{i=1}^{K} h_\theta(x)[i]}{Z'(\theta)}$ and $\frac{\partial}{\partial\theta}\log\sum_{i=1}^{K} h_\theta(x)[i] = \frac{\frac{\partial}{\partial\theta}\sum_{i=1}^{K} h_\theta(x)[i]}{\sum_{i=1}^{K} h_\theta(x)[i]}$.

of Eq. (4) could minimize this KL-divergence either due to Proposition 1. To prove that, we need to recur to Lemma 1 based on [21], which shows how to efficiently compute the gradient of the KL divergence between the real distribution and an approximated distribution modeled via EBMs.

Lemma 1. *Given a training dataset with data $\{x\}$ sampled from distribution $r(x)$, and an energy model distribution $r_\phi(x)$ parameterized by ϕ and associated to an energy function $E_\phi(x)$, the objective of minimizing $D_{KL}(r(x)||r_\phi(x))$ can be optimized by descending the following gradient w.r.t ϕ,*

$$\mathbb{E}_{x^+ \sim r(x)} \left[\frac{\partial E_\phi(x^+)}{\partial \phi} \right] - \mathbb{E}_{x^- \sim r_\phi(x)} \left[\frac{\partial E_\phi(x^-)}{\partial \phi} \right]. \quad (15)$$

The proof can refer [21]. By descending the gradient in Eq. (15), the first term decreases the energy of samples x^+ drawn from the data distribution, while the second term increases the energy of samples x^- drawn from the energy model distribution. Based on Lemma 1, we can introduce following theorem followed by a proof.

Theorem 1. *Let $p(x)$ denote the training data distribution, and $q_\theta(x)$ the energy model distribution represented by the energy function $E'_\theta(x)$ defined in Eq. (6), we can achieve minimization of $D_{KL}(p(x)||q_\theta(x))$ by optimizing Eq. (7).*

Proof. According to Lemma 1, the KL divergence between $p(x)$ and $q_\theta(x)$ can be optimized by descending the gradient in Eq. (15). We can replace the need of computing expectation over the parameterized density $q_\theta(x)$ in Eq. (15) by fixing the parameters θ, denoted by $q_{\bar\theta}(x)$ [4]. As such, Eq. (15) is converted to

$$\mathbb{E}_{p(x)} \left[\frac{\partial E'_\theta(x)}{\partial \theta} \right] - \mathbb{E}_{q_{\bar\theta}(x)} \left[\frac{\partial E'_\theta(x)}{\partial \theta} \right]. \quad (16)$$

Now we can optimize $D_{KL}(p(x)||q_\theta(x))$ via an objective $\mathbb{E}_{p(x)} \left[E'_\theta(x) \right] - \mathbb{E}_{q_{\bar\theta}(x)} \left[E'_\theta(x) \right]$ which holds a numerically same gradient with regard to θ as Eqs. (16) and (15). Then we have[5]

$$\mathbb{E}_{p(x)} \left[E'_\theta(x) \right] - \mathbb{E}_{q_{\bar\theta}(x)} \left[E'_\theta(x) \right]$$
$$= \mathbb{E}_{p(x)} \left[-\log \sum_{i=1}^{K} h_\theta(x)[i] \right] + \mathbb{E}_{q_{\bar\theta}(x)} \left[\log \sum_{i=1}^{K} h_\theta(x)[i] \right]$$
$$\leq \mathbb{E}_{p(x)} \left[-\log h_\theta(x)[y] \right] + \mathbb{E}_{q_{\bar\theta}(x)} \left[\log \sum_{i=1}^{K} h_\theta(x)[i] \right]. \quad (17)$$

---

[4]only the numerical value of Eq. (15) matters in optimization process.

[5]the inequality holds because of the fact $\forall y \leq K$, $\sum_{i=1}^{K} h_\theta(x)[i] \geq h_\theta(x)[y]$.

Therefore, Eq. (7) is an upper bound of an equivalent variant of the KL-divergence between $p(x)$ and $q_\theta(x)$. □

Combing Proposition 1 and Theorem 1, we can conclude that our objective of Eq. (4) can minimize $D_{KL}(p(x)||q_\theta(x))$. Therefore, once our objective converged, we obtain $p(x) \simeq q_\theta(x) = \frac{\exp(-E'_\theta(x))}{Z'(\theta)} \propto \sum_{i=1}^{K} h_\theta(x)[i]$, which result in the summation of K softmax scores of original classes to be directly proportional to the marginal density $p(x)$ and in turn make the $K+1$-th softmax score be negatively correlated to $p(x)$.

4. Experiments

4.1. Experimental Setup

Settings We adopt three settings to evaluate our approach. 1) *Confidence Calibration*: This aims to evaluate the effectiveness of a method in improving confidence calibration. Following [40], we use four datasets: MNIST [24] (MLP), CIFAR-10 [22] (VGG11 [35]), CIFAR-100 [22] (ResNet50 [15]), and Tiny-ImageNet [5] (ResNet50). Network architectures used for these datasets are indicated in the parentheses. 2) *OOD Detection*: A ResNet50 classifier is trained on CIFAR-100 and tested on the combination of CIFAR-100's test split and an OOD dataset, i.e. CIFAR-10 and SVHN [31]. The goal for the classifier is to assign low confidence to as many OOD samples as possible. The accuracy is computed on predictions with confidence higher than a threshold. 3) *Robustness under Corruption*: A classifier is trained on CIFAR-100. Its calibration performance is evaluated on CIFAR-100-C [17], where the images are perturbed by 19 different corruptions with five intensity levels.

Baselines We compare our approach with nine baseline methods: MC-Dropout [7], Temperature Scaling [14], Mixup [37], Label Smoothing [36], TrustScore [20], JEM [12], DBLE [40], and OvA DM [34].

Evaluation Metrics Following prior work [14], we use two metrics to assess how well the confidence in a model's predictions is calibrated, namely Expected Calibration Error (ECE) [29] and Negative Log-Likelihood (NLL). The lower the ECE/NLL, the better the calibration performance. Below we explain in detail how these two metrics are calculated. ECE approximates the expectation of the difference between accuracy and confidence (i.e. the likelihood on the predicted label $\hat{y}$), which can reflect how well the prediction confidence aligns with the true accuracy. Specifically, the confidence estimates made on all test samples are partitioned into L equally spaced bins (following [14, 40, 34], $L = 15$), and the difference between the average confidence

Table 1. Comparison between our approach EOW-Softmax and nine baselines on four benchmark datasets. It is clear that our approach generally leads to a better calibrated model than the baselines (lower ECE & NLL), while maintaining the accuracy. ↑: the higher the better. ↓: the lower the better.

| Method	MNIST (MLP)			CIFAR10 (VGG11)			CIFAR100 (ResNet50)			Tiny-ImageNet (ResNet50)		
	Acc% ↑	ECE% ↓	NLL ↓	Acc% ↑	ECE% ↓	NLL ↓	Acc% ↑	ECE% ↓	NLL ↓	Acc% ↑	ECE% ↓	NLL ↓
Vanilla Training	98.32	1.73	0.29	90.48	6.30	0.43	71.57	19.1	1.58	46.71	25.2	2.95
TrustScore [20]	98.32	2.14	0.26	90.48	5.30	0.40	71.57	10.9	1.43	46.71	19.2	2.75
MC-Dropout [7]	98.32	1.71	0.34	90.48	3.90	0.47	71.57	9.70	1.48	46.72	17.4	3.17
Label Smoothing [36]	98.77	1.68	0.30	90.71	2.70	0.38	71.92	3.30	1.39	**47.19**	5.60	2.93
Mixup [37]	98.83	1.74	0.24	90.59	3.30	0.37	**71.85**	2.90	1.44	46.89	6.80	2.66
JEM [12]	97.23	1.56	0.21	90.36	3.30	0.34	70.28	2.46	1.31	45.97	5.42	2.47
OvA DM [34]	96.67	1.78	0.27	89.56	3.55	0.37	70.11	3.58	1.40	45.55	4.22	2.50
Temperature Scaling [14]	95.14	1.32	0.17	89.83	3.10	0.33	69.84	2.50	1.23	45.03	4.80	2.59
DBLE [40]	98.69	0.97	**0.12**	**90.92**	**1.50**	0.29	71.03	1.10	1.09	46.45	3.60	2.38
EOW-Softmax (*ours*)	**98.91**	**0.88**	0.15	90.24	1.57	**0.25**	71.33	**1.08**	**1.03**	46.97	**3.45**	**2.22**
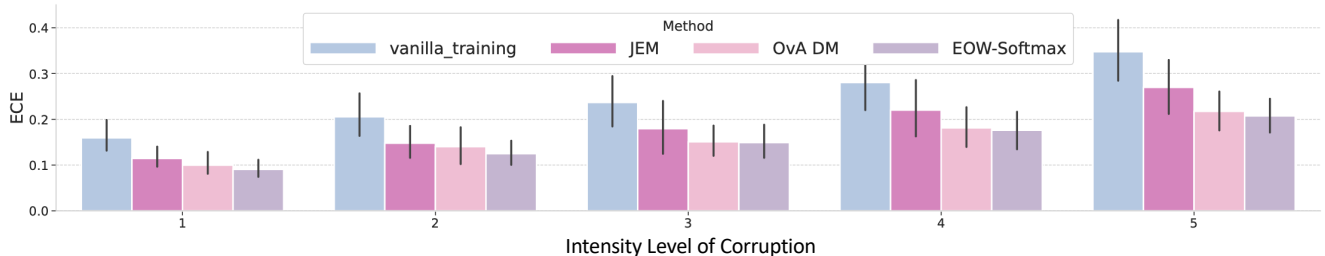
Figure 3. Results on CIFAR-100-C. Each bar represents the mean ECE on 19 different corruptions, with the vertical line segment denoting the standard deviation. In general, EOW-Softmax achieves the lowest ECE under five different corruption levels.

and accuracy within each bin I_l is calculated,

$$\text{ECE} = \sum_{l=1}^{L} \frac{1}{N} | \sum_{x \in I_l} p(\hat{y}|x) - \sum_{x \in I_l} 1(\hat{y} = y)|, \qquad (18)$$

where N denotes the total number of samples in the test set. NLL computes the average negative log-likelihood on all test samples,

$$\text{NLL} = -\frac{1}{N} \sum_{m=1}^{N} \log p(\hat{y}_m | x_m). \qquad (19)$$

Implementation Details We use the SGD optimizer with the learning rate of 1e-4, the momentum of 0.9, and the weight decay of 5e-4. The batch size is set to 64. The number of epochs is 200. The learning rate is decayed by 0.1 at the 100-th and 150-th epoch, respectively. For SGLD, we use a constant step size of 2 and a standard deviation of 1e-3 (see Eq (5)). The number of updates in each SGLD round is set to 100. To ensure that the results are convincing, we run each experiment 5 times with different random seeds and average their results.

4.2. Main Results

Confidence Calibration We first evaluate the calibration performance on four standard datasets, namely MNIST, CIFAR-10/100 and Tiny-ImageNet. The results are shown in Table 1. In general, our approach EOW-Softmax obtains

the best overall calibration performance on most datasets. Comparing with Vanilla Training, we observe that our EOW-Softmax achieves a similar test accuracy, while significantly improves the calibration performance in terms of ECE, especially on the three challenging datasets with natural images—6.30%→1.57% on CIFAR-10, 19.1%→1.08% on CIFAR-100, and 25.2%→3.45% on Tiny-ImageNet. These results strongly suggest that our energy-based open-world uncertainty modeling has great potential for real-world applications as it shows a good balance between test accuracy and calibration performance. JEM and OvA DM are two related methods to ours. JEM is based on joint distribution modeling while OvA DM transforms the conventional softmax classifier to a distance-based one-vs-all classifier. The comparison with these two methods shows that our approach is clearly better in all metrics, which advocates our design of the $K+1$-way softmax for modeling open-world uncertainty. Compared with the top-performing baselines, i.e. Temperature Scaling and DBLE, our EOW-Softmax is highly competitive—it obtains the best ECE on all datasets except CIFAR-10 where the performance is only slightly worse than DBLE. Among these three methods, EOW-Softmax performs the best in maintaining the original test accuracy, whereas Temperature Scaling has to sacrifice the test accuracy in exchange for improvement on ECE. This is because for fair comparison (all methods only have access to the training set), Temperature Scaling has to separate a validation set out from the original training set for

Table 2. Test accuracy on the combination of in-distribution and OOD test set using ResNet50 trained on CIFAR-100.

OOD dataset	Method	Probability threshold			
		0	.25	.5	.75
CIFAR10	Vanilla Training	37.24	42.31	47.67	58.90
	JEM [12]	**42.55**	46.88	0.53	63.21
	OvA DM [34]	39.78	48.54	54.31	65.20
	EOW-Softmax	37.65	**50.11**	**57.32**	**69.00**
SVHN	Vanilla Training	20.24	21.33	24.38	26.90
	JEM [12]	19.87	22.57	26.22	30.75
	OvA DM [34]	20.08	23.99	26.08	30.32
	EOW-Softmax	**20.21**	**25.58**	**28.13**	**32.68**

Table 3. Ablation study on the impact of λ in Eq. (4).

Model	λ	CIFAR10		
		Acc% ↑	ECE% ↓	NLL ↓
VGG11	1	89.81	2.11	0.28
	0.1	**90.24**	**1.57**	**0.25**
	0.01	90.11	3.37	0.36

Model	λ	CIFAR100		
		Acc% ↑	ECE% ↓	NLL ↓
ResNet50	1	70.26	1.33	1.24
	0.1	71.33	**1.08**	**1.03**
	0.01	**71.51**	2.31	1.31

Table 4. Ablation on where to apply the SGLD sampling.

Position	Acc% ↑	ECE% ↓	Second/Iter ↓
Pixel space	73.94	1.89	25.15
Feature stage 1	73.21	2.09	10.31
Feature stage 2	73.48	1.73	5.21
Feature stage 3	73.05	2.17	2.37

tuning its scaling parameter, which reduces the amount of training data.

OOD Detection   We follow [34] to simulate real-world scenarios by training a classifier on CIFAR-100 and testing on the combination of CIFAR-100's test set and an OOD dataset (CIFAR-10/SVHN). The classifier is required to assign low confidence to OOD samples such that their predictions can be rejected by a pre-defined threshold. The results are reported in Table 2 where we compare our approach with JEM and OvA DM (as these two methods are most related to ours), as well as the vanilla training method. The probability threshold is linearly increased from 0 to 0.75. Only predictions with confidence higher than this threshold are kept. From the results, we observe that EOW-Softmax outperforms all baselines with clear margins in different thresholds (except 0). This indicates that a classifier trained with EOW-Softmax suffers much less from the overconfidence problem than the competitors, and is thus safer to be deployed in practical applications.

Robustness under Corruption   We also evaluate our approach on corrupted images, i.e. CIFAR-100-C, and compare with JEM and OVA DM. Specifically, there are five different intensities of corruption as defined in [17], each with 19 different corruption types. Under each intensity level of corruption, we test a model's ECE on images of all corruption types, and report their average result and the standard deviation. The comparison is illustrated in Figure 3. Overall, EOW-Softmax performs favorably against JEM and OVA DM, as well as the vanilla training baseline. Though both EOW-Softmax and JEM are based on energy models for generative modeling, JEM clearly has a larger variation in performance among different corruption types. This is because JEM is more difficult to train—it optimizes the log-likelihood of distribution independently from the classifier learning. In contrast, the 'generative modeling' in EOW-Softmax is seamlessly integrated into the K+1-way classification task, which has a better training stability.

4.3. Ablation study

Hyper-parameter   Recall that λ in Eq. (4) balances between the standard K-way classification loss and the energy-based loss for uncertainty modeling. We experiment with different values (1, 0.1, and 0.01) on CIFAR-10/100 to see the impact of this hyper-parameter. The results in Table 3 show that $\lambda = 0.1$ leads to the best calibration performance (lowest ECE values) on both datasets.

Where to Apply the SGLD Sampling?   In our approach, SGLD sampling is applied to the latent feature space rather than the pixel space as in most EBMs. To justify this design, we experiment on CIFAR-100 with different variants of our approach where the SGLD sampling is applied to different positions, including the pixel space, stage-1, stage-2, and stage-3 (in the neural network). In addition to the test accuracy and the ECE, we also report the training speed (second per iteration) measured using a Tesla K80 GPU. Table 3 shows that applying the SGLD sampling to the pixel space incurs huge computation overhead, while shifting the sampling to the latent space significantly improves the training speed without sacrificing the calibration performance.

5. Conclusion

This paper has addressed the closed-world problem in softmax that causes neural network classifiers to produce overconfident predictions by introducing an energy-based objective to model the open-world uncertainty.

References

[1] Michael Arbel, Liang Zhou, and Arthur Gretton. Generalized energy based models. In *ICLR*, 2021.

[2] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *CVPR*, 2016.

[3] Yoshua Bengio, Grégoire Mesnil, Yann N. Dauphin, and Salah Rifai. Better mixing via deep representations. *CoRR*, abs/1207.4404, 2012.

[4] Tong Che, Ruixiang Zhang, Jascha Sohl-Dickstein, Hugo Larochelle, Liam Paull, Yuan Cao, and Yoshua Bengio. Your gan is secretly an energy-based model and you should use discriminator driven latent sampling. *arXiv preprint arXiv:2003.06060*, 2020.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[6] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In *NeurIPS*, 2019.

[7] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[8] Ruiqi Gao, Yang Song, Ben Poole, Ying Nian Wu, and Diederik P Kingma. Learning energy-based models by diffusion recovery likelihood. In *ICLR*, 2021.

[9] Charles J Geyer. Markov chain monte carlo maximum likelihood. 1991.

[10] Charles J Geyer. Practical markov chain monte carlo. *Statistical science*, pages 473–483, 1992.

[11] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *NeurIPS*, 2014.

[12] 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. *arXiv preprint arXiv:1912.03263*, 2019.

[13] Will Sussman Grathwohl, Jacob Jin Kelly, Milad Hashemi, Mohammad Norouzi, Kevin Swersky, and David Duvenaud. No {mcmc} for me: Amortized sampling for fast and stable training of energy-based models. In *ICLR*, 2021.

[14] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[16] 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 *CVPR*, 2019.

[17] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

[18] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[19] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *CVPR*, 2020.

[20] Heinrich Jiang, Been Kim, Melody Guan, and Maya Gupta. To trust or not to trust a classifier. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[21] Taesup Kim and Yoshua Bengio. Deep directed generative models with energy-based probability estimation. In *ICLR-W*, 2016.

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

[23] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017.

[24] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[25] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

[26] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *ICLR*, 2018.

[27] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, 2018.

[28] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? In *NeurIPS*, 2019.

[29] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[30] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? In *ICLR*, 2019.

[31] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS-W*, 2011.

[32] Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *CoRR*, abs/1412.1897, 2014.

[33] Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. Learning non-convergent non-persistent short-run mcmc toward energy-based model. *arXiv preprint arXiv:1904.09770*, 2019.

[34] Shreyas Padhy, Zachary Nado, Jie Ren, Jeremiah Liu, Jasper Snoek, and Balaji Lakshminarayanan. Revisiting one-vs-all classifiers for predictive uncertainty and out-of-distribution detection in neural networks. *arXiv preprint arXiv:2007.05134*, 2020.

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[36] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[37] Sunil Thulasidasan, Gopinath Chennupati, Jeff Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. *arXiv preprint arXiv:1905.11001*, 2019.

[38] Yoav Wald, Amir Feder, Daniel Greenfeld, and Uri Shalit. On calibration and out-of-domain generalization. *arXiv preprint arXiv:2102.10395*, 2021.

[39] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *ICML*, 2011.

[40] Chen Xing, Sercan Arik, Zizhao Zhang, and Tomas Pfister. Distance-based learning from errors for confidence calibration. In *ICLR*, 2020.

[41] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. In *ICLR*, 2017.

[42] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization in vision: A survey. *arXiv preprint arXiv:2103.02503*, 2021.