Advanced Dropout: A Model-free Methodology for Bayesian Dropout Optimization

Jiyang Xie, *Student Member, IEEE,* Zhanyu Ma, *Senior Member, IEEE,*
Jianjun Lei, *Senior Member, IEEE,* Guoqiang Zhang, *Member, IEEE,*
Jing-Hao Xue, *Member, IEEE,* Zheng-Hua Tan, *Senior Member, IEEE,* and Jun Guo

Abstract—Due to lack of data, overfitting ubiquitously exists in real-world applications of deep neural networks (DNNs). We propose advanced dropout, a model-free methodology, to mitigate overfitting and improve the performance of DNNs. The advanced dropout technique applies a model-free and easily implemented distribution with parametric prior, and adaptively adjusts dropout rate. Specifically, the distribution parameters are optimized by stochastic gradient variational Bayes in order to carry out an end-to-end training. We evaluate the effectiveness of the advanced dropout against nine dropout techniques on seven computer vision datasets (five small-scale datasets and two large-scale datasets) with various base models. The advanced dropout outperforms all the referred techniques on all the datasets.We further compare the effectiveness ratios and find that advanced dropout achieves the highest one on most cases. Next, we conduct a set of analysis of dropout rate characteristics, including convergence of the adaptive dropout rate, the learned distributions of dropout masks, and a comparison with dropout rate generation without an explicit distribution. In addition, the ability of overfitting prevention is evaluated and confirmed. Finally, we extend the application of the advanced dropout to uncertainty inference, network pruning, text classification, and regression. The proposed advanced dropout is also superior to the corresponding referred methods. Codes are available at https://github.com/PRIS-CV/AdvancedDropout.

Index Terms—Deep neural network, dropout, model-free distribution, Bayesian approximation, stochastic gradient variational Bayes.

1 Introduction

Significant progress has been made in deep learning in recent years, especially for computer vision tasks such as image classification [1], [2], [3], [4], [5] and image retrieval [6], [7], [8], [9]. Given that deep neural networks (DNNs) have millions or even billions of parameters, they require a large amount of data for training. However, limited labeled data can be obtained in many applications [4], [10], [11], [12], [13]. Thus, overfitting ubiquitously exists in real-world data analysis, which negatively affects the performance of DNNs.

To address this issue, a dropout technique [14] was proposed to regularize the model parameters by randomly dropping the hidden nodes of DNNs in the training steps to avoid co-adaptations of these nodes. The standard dropout technique and its variants have played an important role in preventing overfitting and popularizing deep learning [15], [16].

Various dropout techniques have been widely utilized in deep neural network training and inference as surveyed in [17]. Existing works [14], [15], [16], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38] differentiate from each other in their use of distributions as shown in Figure 1. Most of the works utilized the Bernoulli distribution for their dropout masks to perform the "dropping" and "holding" in DNNs [14], [20], [21], [22], [23], [24], [25], [26]. In order to regularize model parameters, they randomly drop hidden nodes of DNNs by multiplying Bernoulli distributed masks during training. Meanwhile, some other works applied binary dropout masks, which can be considered as a special Bernoulli distribution in the viewpoint of Bayesian learning [27], [28], [29], [30]. Gaussian distribution is another popular choice for modeling the dropout masks [18], [19], [31], [32]. They considered the Gaussian distribution as a fast approximation of the Bernoulli distribution [31], which was applicable in the local reparameterization trick and better than the Bernoulli distribution [19]. In addition, other distributions including log-normal [16], uniform [18], concrete [33], or beta [15], [34] distributions, were also used in dropout variants.

One issue in common among all the aforementioned dropout techniques is that they all model the dropout masks via model-specific distributions, mainly with Bernoulli distribution [14], [20], [21], [22], [23], [24], [25], [26], [39], and also with Gaussian [18], [19], [31], [32], [40], log-normal [16], uniform [18], concrete [33], or beta [15], [34] distributions. The assumption of a specific distribution introduces a bias that can limit their ability of modeling the dropout masks, as the specific distribution may undesirably restrict the possibility of changes. Few works exceptionally consider whether the distributions are suitable for modelling dropout or not. Soft dropout [15] and β-dropout [34] applied the beta distribution which is able to approximate all the other aforementioned distributions, but it is difficult to optimize

- *J. Xie, Z. Ma, and J. Guo are with the Pattern Recognition and Intelligent Systems Lab., Beijing University of Posts and Telecommunications, China. E-mail: {xiejiyang2013, mazhanyu, guojun}@bupt.edu.cn*
- *J. Lei is with the School of Electrical and Information Engineering, Tianjin University, Tianjin, China. E-mail: jjlei@tju.edu.cn*
- *G. Zhang is with the School of Electrical and Data Engineering, University of Technology Sydney, Australia. E-mail: guoqiang.zhang@uts.edu.au*
- *J. -H. Xue is with the Department of Statistical Science, University College London, United Kingdom. E-mail: jinghao.xue@ucl.ac.uk*
- *Z. -H. Tan is with the Department of Electronic Systems, Aalborg University, Denmark. E-mail: zt@es.aau.dk*

(Corresponding author: Zhanyu Ma)

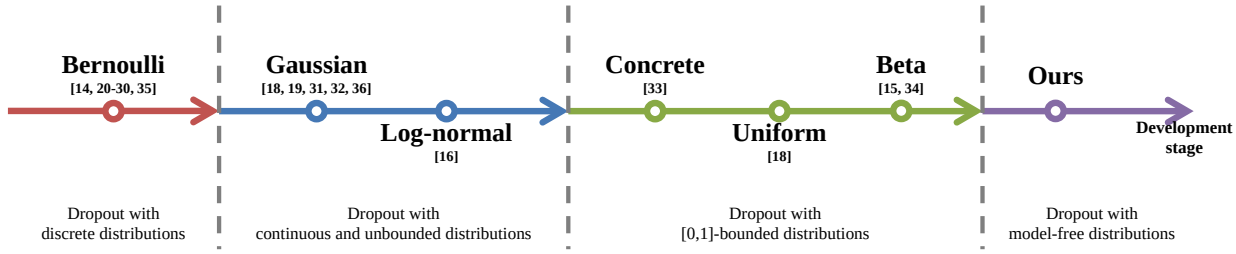


Fig. 1: Development of the distributions applied in various dropout techniques.

the parameters of the beta distribution for dropout masks. One solution to this problem is Bayesian approximation using tractable distributions [15] and manual selection of beta distribution parameters [34], which both have limitations for dropout training.

In this paper, we propose a model-free methodology for dropout, named advanced dropout, to further improve the capability of overfitting prevention and boost the performance of DNNs. The dropout technique applies a model-free and easily implemented distribution with a parametric prior to adaptively adjust the dropout rate. Furthermore, the prior parameters are optimized by the stochastic gradient variational Bayes (SGVB) inference [41] to perform an end-to-end training procedure of DNNs. Our major contributions can be summarized as follows:

- We introduce a novel *model-free* methodology that is more flexible than the existing techniques with specifically explicit distributions in dropouts. By choosing proper parameters, the proposed methodology is able to replace all the aforementioned distributions to generalize the other dropout variants under the model-free framework (Section 4.1).
- We apply a parametric prior form for the model-free distribution's parameters to *adaptively adjust the dropout rate* based on input features. The prior can be integrated into the end-to-end training such that the calculation is facilitated (Section 4.2).
- We propose *advanced dropout* consisting of the two key components, *i.e.*, a model-free distribution for dropout masks and a parametric prior for the parameters of the distribution. We evaluate the effectiveness of the advanced dropout with image classification task. Experimental results demonstrate that the advanced dropout outperforms all the nine recently proposed techniques on seven widely used datasets with various base models. We further compare training time and effectiveness ratios and find that the advanced dropout achieves highest effectiveness ratios on most of the datasets (Section 5.1).
- We extensively study the advanced dropout technique on the following aspects: the effectiveness of each component, dropout rate characteristics, and the ability of overfitting prevention. We clarify that the key components are essential for the technique (Section 5.2). Meanwhile, the analysis of dropout rate characteristics demonstrates that the technique is able to achieve a *stable convergence* of dropout rate (Section 5.3.1), illustrates the learned distributions of dropout masks (Section 5.3.2), and shows its better performance than the one using dropout rate generation without an explicit distribution (Section 5.3.3). Finally, *superior ability of preventing overfitting* of the advanced dropout technique is shown (Section 5.4).
- We extend the application of the advanced dropout technique to *uncertainty inference*, *network pruning*, *text classification*, and *regression*. We conduct several experiments on the model uncertainty inference and show the improvement of the advanced dropout technique (Section 6.1). We also employ the advanced dropout technique as a network pruning technique and compare it with the state-of-the-art methods to show the performance improvement (Section 6.2). In addition, we apply the advanced dropout technique on text classification (Section 6.3) and regression (Section 6.4), and find its superiority among all the referred dropout techniques.

2 Related Work

After Hinton *et al*. [14] introduced the standard dropout in 2012, many variants of dropout have been proposed in recent years, as the significant effectiveness of dropout on preventing overfitting has been discovered when applied on deep and wide DNN structures. Distinct distributions were applied by the dropout variants to their own design strategies of DNN regularization. In particular, six distributions were introduced, including Bernoulli [14], [20], [21], [22], [23], [24], [25], [26], [39], Gaussian [18], [19], [31], [32], [40], log-normal [16], uniform [18], concrete [33], and beta [15], [34] distributions. As shown in Figure 1, the dropout techniques can be divided into four development stages with their own kinds of distributions. The following parts of this section will review the works with different distributions in the stages, respectively.

2.1 On Discrete Distributions

In the research of dropouts, the discrete distributions commonly refer to the Bernoulli distribution and the binary distribution, while the latter can be considered as a special Bernoulli distribution in the viewpoint of Bayesian learning.

Most of works utilized the Bernoulli distribution for their dropout masks to perform "dropping" and "holding" in DNNs. This standard dropout aims to regularize the model parameters and reduce overfitting by randomly dropping hidden nodes of an fully connected (FC) neural network with Bernoulli distributed masks during training [14]. DropConnect [20] randomly selected a subset of weights within the network to zero, rather than activations in each layer. Meanwhile, standout [22] performed as a binary belief network, was trained jointly with the DNN using stochastic gradient descent (SGD), and computed the local expectations of binary dropout variables. Maeda [23] introduced

a Bayesian interpretation to optimize the dropout rate, which was beneficial for model training and prediction. Gal and Ghahramani [24] utilized standard dropout to predict model uncertainty in DNNs in regression, classification, and reinforcement learning. All the aforementioned techniques rely on standard dropout with fixed parameters which are empirically set. Later, a dropout variant has been proposed for RNNs focusing on time dependence representation and demonstrated outstanding effectiveness [25]. Spectral dropout [26] instead implemented standard dropout on the spectrum dimension of the convolutional feature maps, preventing overfitting by eliminating the weak Fourier domain coefficients of activations. Jumpout [21] sampled the dropout rate from a monotone decreasing distribution and adaptively normalized the rate at each layer to keep the effective rate. Wang *et al*. [39] proposed a lightweight complexity algorithm called Rademacher Dropout (Rad-Dropout) to achieve adaptive adjustment of dropout rates.

Furthermore, some works applied the binary distribution for the dropout masks, and selected or dropped nodes according to some specific rules. Ko *et al*. [27] introduced controlled dropout and intentionally chose the activations to drop non-randomly for improving the training speed and the memory efficiency. Alpha-divergence dropout [28] applied the alpha divergence as a regularization term, replacing the conventional Kullback-Leibler (KL) divergence in approximate variational inference for dropout training. Ising-dropout [29] dropped the activations in a DNN using Ising energy of the network to eliminate the optimization of unnecessary parameters during training. In addition, Chen *et al*. [30] proposed mutual information-based dropout (DropMI), introducing mutual information dynamic analysis to the model and highlighting the important activations that are beneficial to the feature representation.

However, one issue with the discrete distributions is the difficulty of dropout rate optimization combined in DNN training, which does not exist with continuous distributions. Except the fixed dropout rate in [14], [20], [24], [26], some works addressed the issue by updating asynchronously [22], adding regularization terms into the loss functions [23], and randomly selecting from other distributions [21].

2.2 On Continuous and Unbounded Distributions

The works in [16], [18], [19], [31], [32] introduced the continuous and unbounded distributions to dropouts, mainly including the Gaussian and the log-normal distributions. They took advantages of the continuity and the differentiability of the distributions for end-to-end training of DNNs.

The Gaussian distribution is another popular choice for the dropout masks. It is considered a fast approximation of the Bernoulli distribution [31] and is applicable in the local reparameterization trick, better than the Bernoulli distribution [19]. Wang and Manning [31] proposed a Gaussian approximation of standard dropout under the variational Bayes framework, called fast dropout, with virtually identical regularization performance but much faster convergence. Another extension of dropout in [32] applied Gaussian multiplicative noise with unit mean, replacing the Bernoulli noise. It can be interpreted as a variational method given a particular prior over the network weights to some extent [32]. Kingma et.al. [19] introduced variational dropout where the dropout rate was optimized by the stochastic gradient variational Bayes (SGVB) inference [41]. Continuous dropout [18] replaced the Bernoulli distribution by a Gaussian distribution with mean 0.5 or a uniform one as the prior of continuous masks in practice, even though the variance of the Gaussian distribution could not be optimized during training. Liu *et al*. [40] proposed a variational Bayesian dropout with a hierarchical prior.

In addition, information dropout [16] improved dropout by information theory principles and adapted the dropout rate to the data automatically under the Bayesian theory. It applied the log-normal distribution, which is also an unbounded-domain distribution as the Gaussian distribution, and obtained positive dropout masks.

However, masks with large values approaching infinite can be sampled and negatively affect gradient backpropagation, resulting in gradient exploding, which is a huge problem in practice.

2.3 On $[0, 1]$-bounded Distributions

In addition, other distributions with $[0, 1]$ bound were also used in the dropout variants. The $[0, 1]$-bounded distributions contain the concrete, the uniform, and the beta distributions, which are also continuous distributions.

Gal *et al*. [33] proposed concrete dropout by introducing a concrete distribution for the dropout masks. The concrete distribution is the first $[0, 1]$-bounded distribution applied to dropout and obtained remarkable improvement on overfitting prevention. Furthermore, the uniform distribution was also applied by the continuous dropout and compared with the Gaussian distribution with different variance settings [18]. The uniform distribution usually performed worse than the Gaussian distribution according to the experimental results in [18], due to the low degree of freedom.

Given that beta distribution with different parameters can approximate the Bernoulli, the Gaussian, and the uniform distributions to some extent [34], it should be great potential than other distributions in dropout regularization. The β-dropout technique performed best over others and conducted finer control of its regularization; however, the parameters of the beta distributed masks have to be manually selected [34]. In soft dropout [15], the soft dropout masks were expressed by beta distributed variables which have better continuity than the Bernoulli distribution and higher flexibility in shape than the Gaussian distribution. However, the beta distribution is unfeasible to be directly extended to the SGVB inference which is one of the most effective solutions combining variational inference with the SGD optimization [15]. To utilize soft dropout in the SGVB inference, the beta distribution was approximated by half-Gaussian and half-Laplace distributions, respectively, and the technique could adaptively adjust its dropout rate in the training process [15]. However, the half-Gaussian and half-Laplace approximations are complex and can only approximate part of shapes (U shape) well, but hardly represent other shapes (*e.g.*, uniform shape). Furthermore, the parameters of these two approximations are optimized without following any prior distributions, which affects the performance of overfitting prevention.

In this context, we consider to design a more generalized and model-free distribution followed by the dropout masks,

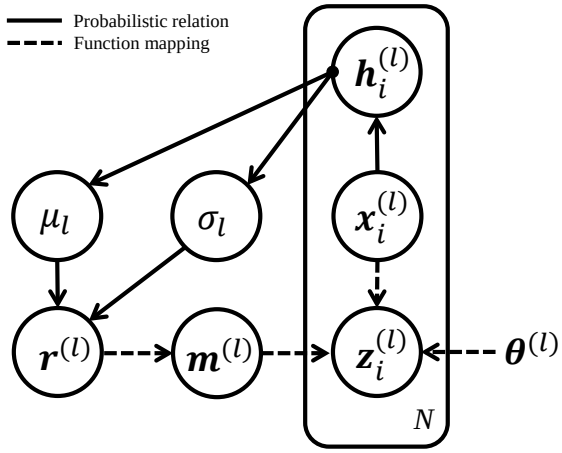


Fig. 2: Probabilistic graphical model of the advanced dropout technique. Two key components in the architecture are 1) the model-free distribution $g(\boldsymbol{m}^{(l)}|\mu_l, \sigma_l)$ and 2) the parametric prior $p(\mu_l, \sigma_l|\boldsymbol{h}_1^{(l)}, \cdots, \boldsymbol{h}_N^{(l)}) \prod_{i=1}^{N} p(\boldsymbol{h}_i^{(l)}|\boldsymbol{x}_i^{(l)})$.

which is able to approximate the widely used Bernoulli, Gaussian, uniform, concrete, and beta distributions to generalize the fundamental principles of all the dropout variants and has an easy implementation in the SGVB inference for optimization. Although all the aforementioned techniques introduced exact distributions to explicitly explain the dropout mask, we utilize the model-free distributions and overcome the difficulties of them (*i.e.*, shape limitations and distribution parameter optimization) discussed above. In addition, a suitable prior is essential in Bayesian learning. We introduce a parametric prior which is supported by features and integrate information to better optimize the parameters of the model-free distribution.

3 Preliminaries

We start with an L-layer DNN, which has $K_l, l = 1, \cdots, L$, features in the lth layer. The parameter set of the DNN is defined as $\boldsymbol{\Theta} = \{\boldsymbol{\theta}^{(l)}\}_{l=1}^{L}$, where $\boldsymbol{\theta}^{(l)} \in R^{K_l \times K_{l-1}}$ is the network parameter matrix of the lth layer. For each layer, we define a layer model with standard dropout [14] as

$$\boldsymbol{z}^{(l)} = a\left(\boldsymbol{m}^{(l)} \odot (\boldsymbol{\theta}^{(l)}\boldsymbol{x}^{(l)})\right), \tag{1}$$

where $\boldsymbol{x}^{(l)}$ and $\boldsymbol{z}^{(l)}$, with $\boldsymbol{x}^{(l+1)} = \boldsymbol{z}^{(l)}$ for $1 \leq l < L$, are the input and the output of the layer, respectively. $a(\cdot)$ is the activation function, *e.g.*, rectified linear unit (ReLU) function, $\odot$ is the Hadamard product operation, and $\boldsymbol{m}^{(l)}$ is the dropout mask vector.

In the standard dropout framework [14], the elements in $\boldsymbol{m}^{(l)}$ are random variables following *i.i.d.* Bernoulli distributions with the dropout rate $\rho_l = 1 - \mathrm{E}[m_j^{(l)}] = 1 - p_l, j = 1, \cdots, K_l$, as

$$\boldsymbol{m}^{(l)} \sim \prod_{j=1}^{K_l} \mathrm{Bernoulli}_j(p_l) = \prod_{j=1}^{K_l} p_l^{m_j^{(l)}} (1 - p_l)^{1 - m_j^{(l)}}, \tag{2}$$

where p_l is the parameter of the Bernoulli distribution. In the training step, the dropout mask vector $\boldsymbol{m}^{(l)}$ is sampled from its distribution, producing a binary mask vector. Meanwhile, $\boldsymbol{x}^{(l)}$ is scaled by the mean vector $[p_l]_{K_l}$ in the test step [14].

In order to better express the continuity of the dropout masks, the Bernoulli distribution is replaced by different distributions including Gaussian [32], log-normal [16], uniform [18], concrete [33], and beta [15] distributions. Considering the recent soft dropout [15] as an example, the prior distribution of $\boldsymbol{m}^{(l)}$ is modified into a multi-dimensional beta distribution, which can be considered as a product of *i.i.d.* beta distributions with the dropout rate $\rho_l = 1 - \frac{\alpha_l}{\alpha_l + \beta_l}$ as

$$\begin{aligned} \boldsymbol{m}^{(l)} &\sim \prod_{j=1}^{K_l} \mathrm{Beta}_j(\alpha_l, \beta_l) \\ &= \prod_{j=1}^{K_l} \frac{(m_j^{(l)})^{\alpha_l - 1}(1 - m_j^{(l)})^{\beta_l - 1}}{\int_0^1 u^{\alpha_l - 1}(1-u)^{\beta_l - 1}\, du}, \end{aligned} \tag{3}$$

where $\alpha_l, \beta_l > 0$ are the shape parameters. Similar to standard dropout, $\boldsymbol{x}^{(l)}$ is scaled by the mean vector $[\frac{\alpha_l}{\alpha_l + \beta_l}]_{K_l}$ in the test step.

By replacing the binary dropout masks with the beta distributed variables, the soft dropout masks are continuously distributed in the interval $[0, 1]$, rather than only zero or one. Thus, it samples the masks from an infinite space for parameter selection, giving infinite states of each soft dropout mask. The binary mask optimization space of the standard dropout [14] can be considered as a subset of the soft one.

4 Advanced Dropout

In this section, an advanced dropout technique is proposed for DNNs. The probabilistic graphical model is shown in Figure 2, including two key components in the architecture, *i.e.*, the model-free distribution and the parametric prior. After introducing the two parts in Section 4.1 and Section 4.2, respectively, we discuss how to optimize the whole advanced dropout technique by the SGVB inference [41] in DNN training within the SGD algorithm in Section 4.3.

4.1 Model-free Distribution for Advanced Dropout

The development of the distributions applied in various dropout techniques is illustrated in Figure 1. The dropout masks firstly followed the Bernoulli distribution, a discrete distribution, to perform the fully "dropping" and "holding" in DNNs [14], [20], [21], [22], [23], [24], [25], [26], [39]. Then, the distribution was replaced by the Gaussian [18], [19], [31], [32], [40] or the log-normal [16] distributions, which are continuous and unbounded distributions. These distributions, as the approximations of the Bernoulli distribution, performed well. However, masks with large values approaching infinity can be sampled to some extent, leading to gradient exploding, which can be a huge problem in practice.

To address this problem, $[0, 1]$-bounded distributions including the concrete [33], the uniform [18], and the beta [15], [34] distributions were introduced and achieved better performance. In addition, the asymmetry of a distribution can introduce more flexible shapes of the probability density function (PDF) for the distribution and adapt different dropout rates in DNN training, which are beneficial to the dropout techniques. As we know that a symmetric distribution can merely present dropout with its dropout rate at 0.5 (*e.g.*, the Gaussian distribution with mean 0.5 and the uniform distribution in [18]), an asymmetric one can express not only the half-dropping case, but also all the other cases of the dropout rate, implemented by different values of the parameters.

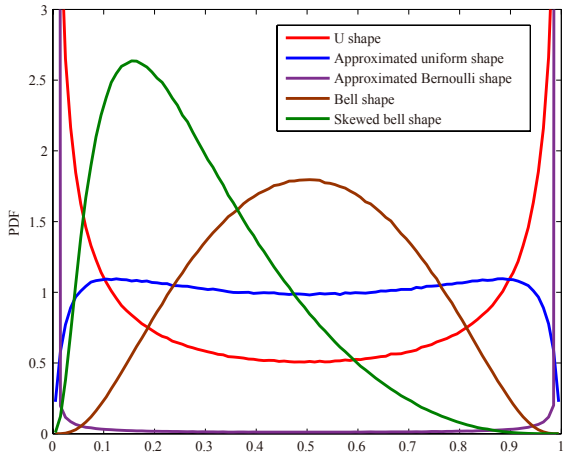


Fig. 3: PDFs of the model-free distribution g applied in experiments with different parameter settings. The parameter settings of g with different shapes are $\mu = 0, \sigma = 3$ for the "U" shape, $\mu = 0, \sigma = 1.6$ for the approximated uniform shape, $\mu = 0, \sigma = 150$ for the approximated Bernoulli shape, $\mu = 0, \sigma = 1$ for the bell shape, and $\mu = -1, \sigma = 1$ for the skewed bell shape. Note that we can approximate the Gaussian distribution by the bell-shaped g and the log-normal distribution by the skewed bell-shaped g, and the beta distribution by the U-, bell-, skewed bell-shaped g, respectively.

In addition, shape is significant for the dropout distribution, because different shapes (*e.g.*, "U" and bell shapes) can represent distinct states of the masks, mediately reflecting the state of the model. The beta distribution has different shapes of PDF, including the "U", the bell, the skewed bell, and the uniform shapes. However, the concrete distribution has only the "U" shape in practice. Thus, the beta distribution is preferred to the other distributions mentioned above. Although all the aforementioned dropout techniques have their own distributions, *do these distributions suit the needs of dropout itself?*

To further address the issue, we introduce a model-free distribution which satisfies *all* the properties of the aforementioned distributions and *all* the requirements of soft dropout. The model-free distribution will not be restricted by any distribution forms and can prevent soft dropout from the infeasibility incurred by some distributions, such as the beta distribution, in the end-to-end training [15]. It can be defined in arbitrary forms and we are able to produce the various shapes by adjusting the parameters.

Here, we propose a model-free distribution, which is denoted by g. For the jth element of the dropout mask $m_j^{(l)} \sim g(m_j^{(l)})$ in the lth layer of the DNN, another hidden variable $r_j^{(l)}$ is introduced as the seed variable, following a seed distribution $s(\cdot)$. Then, by introducing a monotonic and differentiable function $k(\cdot)$ as the mapping function, which has its value space in the interval of $[0, 1]$, we obtain the $[0, 1]$-bounded continuous variable $m_j^{(l)} = k(r_j^{(l)})$ directly. In this case, the seed distribution $s(\cdot)$ and the mapping function $k(\cdot)$ should satisfy two conditions:

1) $s(\cdot)$ can be transformed into a differentiable function of its parameters and standard distribution, and should be easy to be sampled. For example, Gaussian, Laplace, exponential, or uniform distributions.
2) $k(\cdot)$ should be monotonic and differentiable in its domain, and the domain of output values of $k(\cdot)$ should be in the interval of $[0, 1]$. For example, Sigmoid function and some piecewise differentiable functions.

In principle, *any* seed distribution and mapping function pair can be applied to construct the model-free distribution, as long as they satisfy the conditions 1) and 2), respectively. The model-free distribution g can be *any* form satisfying $\int g(x) d\,x = 1$. The relationship between $s(\cdot)$, $k(\cdot)$, and g can be obtained via the formula for calculating the distribution of functions of random variables as

$$g(m_j^{(l)}) = s(k^{-1}(m_j^{(l)})) \frac{d\,k^{-1}(m_j^{(l)})}{d\,m_j^{(l)}}, \tag{4}$$

where $k^{-1}(\cdot)$ is the inverse function of $k(\cdot)$. Note that $k(\cdot)$ is a monotonic function, so that $k^{-1}(\cdot)$ can be always obtained.

As we know in DNNs, the Gaussian distribution is a popular assumption and the Sigmoid function is widely used, which satisfy the conditions of $s(\cdot)$ and $k(\cdot)$, respectively. We involve them into the model-free distribution framework. The linear additivity of the Gaussian distribution and differentiability of the Sigmoid function make the model-free distribution g feasible for the SGVB inference. In addition, $k(\cdot)$ allows $m_j^{(l)}$ falling in the bounded interval $[0, 1]$. In this case, the variable $m_j^{(l)}$ can be explicitly defined by $r_j^{(l)}$ as

$$\begin{aligned} m_j^{(l)} &= k(r_j^{(l)}) = \text{Sigmoid}(r_j^{(l)}) = \frac{1}{1 + e^{-r_j^{(l)}}}, \\ r_j^{(l)} &\sim \mathcal{N}(\mu_l, \sigma_l^2), \end{aligned} \tag{5}$$

where μ_l and σ_l are the mean and the standard deviation of the seed variable $r_j^{(l)}$. Furthermore, the PDF of g in this case can be defined as

$$\begin{aligned} g(m_j^{(l)}) &= \mathcal{N}(k^{-1}(m_j^{(l)}); \mu_l, \sigma_l^2) \frac{d\,k^{-1}(m_j^{(l)})}{d\,m_j^{(l)}} \\ &= \frac{1}{\sqrt{2\pi}\sigma_l} e^{-\frac{\left(\ln m_j^{(l)} - \ln(1 - m_j^{(l)}) - \mu_l\right)^2}{2\sigma_l^2}} \left(\frac{1}{m_j^{(l)}(1 - m_j^{(l)})} \right), \end{aligned} \tag{6}$$

where $m_j^{(l)}$ is subject to the interval of $[0, 1]$ by $k(\cdot)$. Here, although any seed distribution and mapping function pairs satisfying the conditions 1) and 2) can be introduced into the model-free distribution, we use the Gaussian distribution as the seed distribution for the purpose of easy implementation. It is worth to mention that the PDF of g in (6) has the same form with that of logit-normal distribution. The logit-normal distribution can be considered as a special case of the model-free distribution. We introduce the logit-normal distributed variables generated by Gaussian variables through the Sigmoid function to exhibit the effectiveness of the proposed methodology in experiments merely.

The expectation of the variable $m_j^{(l)}$ is then calculated as

$$\begin{aligned} \mathrm{E}\left[m_j^{(l)}\right] &= \int_0^1 m_j^{(l)} g(m_j^{(l)}) d\,m_j^{(l)} \\ &= \int_{-\infty}^{+\infty} \text{Sigmoid}(r_j^{(l)}) \mathcal{N}(r_j^{(l)}; \mu_l, \sigma_l^2) d\,r_j^{(l)} \\ &\approx \int_{-\infty}^{+\infty} \phi(\sqrt{\frac{\pi}{8}} r_j^{(l)}) \mathcal{N}(r_j^{(l)}; \mu_l, \sigma_l^2) d\,r_j^{(l)} \\ &= \phi\left(\frac{\mu_l}{\sqrt{\frac{8}{\pi} + \sigma_l^2}} \right) \approx \text{Sigmoid}\left(\frac{\mu_l}{\sqrt{1 + \frac{\pi}{8}\sigma_l^2}} \right), \end{aligned} \tag{7}$$

where

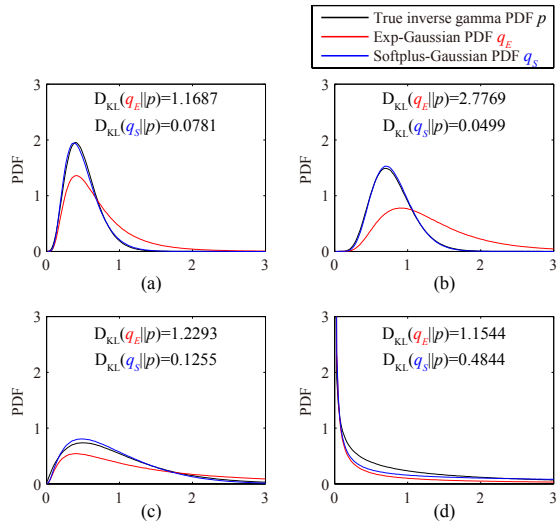


Fig. 4: PDFs of the softplus-Gaussian distribution (in blue) and log-normal distribution [41] (in red) approximating the inverse gamma distribution (in black) with different parameters: (a) $k = 5, \theta = 0.1$; (b) $k = 8, \theta = 0.1$; (c) $k = 2, \theta = 0.5$; (d) $k = 0.5, \theta = 3$.

$$\phi(x) = \int_{-\infty}^{x} \mathcal{N}(t; 0, 1)dt \approx \text{Sigmoid}\left(\sqrt{\frac{8}{\pi}}x\right) \tag{8}$$

is the cumulative distribution function of a standard normal distribution [42].

Finally, we discuss the distinct shape advantages of the model-free distribution utilized in this paper. As shown in Figure 3, the model-free distribution g (with the Gaussian seed distribution and the Sigmoid function) has various shapes corresponding to different parameter settings, including "U", bell, skewed bell, and approximated uniform and Bernoulli shapes. It can approximate the Bernoulli, the uniform, the Gaussian, the log-normal, and the beta distributions by adjusting μ_l and σ_l. Meanwhile, the concrete distribution is a special case in the form of the model-free PDF g. Therefore, the model-free distribution can replace all the aforementioned distributions applied in dropout variants.

4.2 Parametric Prior Distribution for Parameters of Model-free Distribution

After defining the model-free distribution form of the dropout masks $\boldsymbol{m}^{(l)}$ in Figure 2, a prior distribution of the distribution parameters μ_l and σ_l is required in the Bayesian inference. We know that the prior of μ_l and σ_l given the input features $\boldsymbol{x}^{(l)}$ of the lth layer can integrate information and extract features for helping the dropout technique learning a better distribution [16], [22], which means that we can adaptively adjust the dropout rate by optimizing distribution parameters with the prior. In this case, we introduce the prior distribution $p(\mu_l, \sigma_l|\boldsymbol{x}^{(l)})$ as

$$p(\mu_l, \sigma_l|\boldsymbol{x}^{(l)}) = \int p_\mu(\mu_l|\boldsymbol{h}^{(l)})p_\sigma(\sigma_l|\boldsymbol{h}^{(l)})p_h(\boldsymbol{h}^{(l)}|\boldsymbol{x}^{(l)})d\,\boldsymbol{h}^{(l)}, \tag{9}$$

where $\boldsymbol{h}^{(l)}$ is the multivariate Gaussian distributed hidden states of the lth layer.

To simplify Bayesian optimization later on, we assume μ_l and σ_l follow Gaussian and inverse gamma distributions, respectively. That is, we set their prior distributions as

$$\mu_l \sim \mathcal{N}(\mu_l|\cdot) \approx p_\mu(\mu_l|\boldsymbol{h}^{(l)}), \tag{10}$$

$$\sigma_l \sim \mathcal{IG}(\sigma_l|\cdot) \approx p_\sigma(\sigma_l|\boldsymbol{h}^{(l)}), \tag{11}$$

$$\boldsymbol{h}^{(l)} \sim \mathcal{N}(\boldsymbol{h}^{(l)}|\cdot) \approx p_h(\boldsymbol{h}^{(l)}|\boldsymbol{x}^{(l)}), \tag{12}$$

where $\mathcal{IG}$ is the inverse gamma distribution. Here, p_μ and p_σ perform as encoders, and the maximization of the distributions $p_\mu(\mu_l|\boldsymbol{h}^{(l)})$ and $p_\sigma(\sigma_l|\boldsymbol{h}^{(l)})$ can be approximated by multi-layer perceptrons (MLPs) with Gaussian outputs [41] as

$$\begin{aligned}\hat{\mu}_l &= \underset{\mu_l}{\text{argmax}} \prod_{i=1}^{N} p_\mu(\mu_l|\boldsymbol{h}_i^{(l)}) \\ &\approx \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\Omega}_\mu^{(l)}\boldsymbol{h}_i^{(l)} + \boldsymbol{b}_\mu^{(l)},\end{aligned} \tag{13}$$

$$\begin{aligned}\hat{\sigma}_l &= \underset{\sigma_l}{\text{argmax}} \prod_{i=1}^{N} p_\sigma(\sigma_l|\boldsymbol{h}_i^{(l)}) \\ &\approx \frac{1}{N}\sum_{i=1}^{N} \text{Softplus}\left(\boldsymbol{\Omega}_\sigma^{(l)}\boldsymbol{h}_i^{(l)} + \boldsymbol{b}_\sigma^{(l)}\right),\end{aligned} \tag{14}$$

where $\boldsymbol{\gamma}^{(l)} = \{\boldsymbol{\Omega}_\mu^{(l)}, \boldsymbol{\Omega}_\sigma^{(l)}, \boldsymbol{b}_\mu^{(l)}, \boldsymbol{b}_\sigma^{(l)}\}$ are the weights and the biases of the MLPs, N is sample number, and the softplus function $\text{Softplus}(\cdot)$ is defined as

$$\text{Softplus}(x) = \ln(1 + e^x). \tag{15}$$

Here, we define the inverse gamma approximation by the softplus function with Gaussian input as softplus-Gaussian distribution, which is a normalized distribution. To evaluate the effectiveness of the softplus-Gaussian distribution, we conduct a group of experiments in comparison with the classical log-normal distribution used in [41], as shown in Figure 4. The distributions approximate the true inverse gamma approximation by moment matching. The subfigures in Figure 4 show that the proposed softplus-Gaussian distribution is better than the referred one, due to smaller KL divergences in all the different cases.

In addition, the optimal hidden state $\hat{\boldsymbol{h}}^{(l)}$ (by maximizing $p_h(\boldsymbol{h}^{(l)}|\boldsymbol{x}^{(l)})$) can be inferred as

$$\begin{aligned}\hat{\boldsymbol{h}}^{(l)} &= \underset{\boldsymbol{h}^{(l)}}{\text{argmax}}\, p_h(\boldsymbol{h}^{(l)}|\boldsymbol{x}^{(l)}) \\ &\approx \boldsymbol{\Omega}_h^{(l)}\boldsymbol{x}^{(l)} + \boldsymbol{b}_h^{(l)},\end{aligned} \tag{16}$$

where $\boldsymbol{\delta}^{(l)} = \{\boldsymbol{\Omega}_h^{(l)}, \boldsymbol{b}_h^{(l)}\}$ are the weights and the biases of the MLP.

4.3 Stochastic Gradient Variational Bayes (SGVB) for Advanced Dropout

After constructing the whole advanced dropout technique as shown via the probabilistic graphical model in Figure 2, we consider how to optimize it using the SGVB inference [41].

We first define the dataset $\boldsymbol{D} = \{\boldsymbol{X}, \boldsymbol{Y}\}$ for DNN training where $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^N$ and $\boldsymbol{Y} = \{\boldsymbol{y}_i\}_{i=1}^N$ are sets of input and target with N samples, respectively, and the dropout-masked parameter set $\boldsymbol{W} = \{\boldsymbol{w}^{(l)}\}_{l=1}^L$ in which

$$\boldsymbol{w}^{(l)} = \text{diag}(\boldsymbol{m}^{(l)})\boldsymbol{\theta}^{(l)}, \tag{17}$$

where $\text{diag}(\cdot)$ is the matrix operation that transforms a vector into a squared diagonal matrix with the vector as the main diagonal. In addition, we define the parameter set $\boldsymbol{\Lambda} = \{\boldsymbol{\lambda}_l\}_{l=1}^L$ where $\boldsymbol{\lambda}_l = \{\mu_l, \sigma_l, \boldsymbol{\gamma}^{(l)}, \boldsymbol{\delta}^{(l)}\}$. The joint parameter set including original parameters $\boldsymbol{\Theta}$ and $\boldsymbol{\Lambda}$ is defined as $\boldsymbol{\Phi} = \{\boldsymbol{\Theta}, \boldsymbol{\Lambda}\}$.

We can then obtain the joint distribution of the dataset $\boldsymbol{D}$ and the dropout-masked parameter set $\boldsymbol{W}$ as

$$p(\boldsymbol{D}, \boldsymbol{W}) = p(\boldsymbol{D}|\boldsymbol{W})p(\boldsymbol{W}). \tag{18}$$

Introducing the approximated distribution $q_{\boldsymbol{\Phi}}(\boldsymbol{W}|\boldsymbol{Z})$ of $\boldsymbol{W}$ given $\boldsymbol{Z} = \{\boldsymbol{z}^{(0)}, \cdots, \boldsymbol{z}^{(L-1)}\}$ ($\boldsymbol{z}^{(l)} = \boldsymbol{x}^{(l+1)}$) as

$$q_{\boldsymbol{\Phi}}(\boldsymbol{W}|\boldsymbol{Z}) = \prod_l q_{\boldsymbol{\Phi}}(\boldsymbol{w}^{(l)}|\boldsymbol{z}^{(l-1)}), \tag{19}$$

we divide the left-hand side (LHS) and the right-hand side (RHS) of (18) by $q_{\boldsymbol{\Phi}}(\boldsymbol{W}|\boldsymbol{Z})$ and take the logarithms of each side which give

$$\log \frac{p(\boldsymbol{D}, \boldsymbol{W})}{q_{\boldsymbol{\Phi}}(\boldsymbol{W}|\boldsymbol{Z})} = \log \frac{p(\boldsymbol{D}|\boldsymbol{W})p(\boldsymbol{W})}{q_{\boldsymbol{\Phi}}(\boldsymbol{W}|\boldsymbol{Z})}. \tag{20}$$

We consider the expectation of the LHS and the RHS of (20), *w.r.t.* $q_{\boldsymbol{\Phi}}(\boldsymbol{W}|\boldsymbol{Z})$ as

$$\begin{aligned} &\underbrace{\int q_{\boldsymbol{\Phi}}(\boldsymbol{W}|\boldsymbol{Z}) \log \frac{p(\boldsymbol{D}, \boldsymbol{W})}{q_{\boldsymbol{\Phi}}(\boldsymbol{W}|\boldsymbol{Z})} \, d\boldsymbol{W}}_{L(\boldsymbol{\Phi})} \\ &= \underbrace{\int q_{\boldsymbol{\Phi}}(\boldsymbol{W}|\boldsymbol{Z}) \log p(\boldsymbol{D}|\boldsymbol{W}) \, d\boldsymbol{W}}_{L_D(\boldsymbol{\Phi})} - \underbrace{\int q_{\boldsymbol{\Phi}}(\boldsymbol{W}|\boldsymbol{Z}) \log \frac{q_{\boldsymbol{\Phi}}(\boldsymbol{W}|\boldsymbol{Z})}{p(\boldsymbol{W})} \, d\boldsymbol{W}}_{\mathrm{D_{KL}}(q_{\boldsymbol{\Phi}}(\boldsymbol{W}|\boldsymbol{Z})||p(\boldsymbol{W}))}, \end{aligned} \tag{21}$$

where the LHS $L(\boldsymbol{\Phi})$ is the lower bound of the expectation of posterior distribution $\mathrm{E}_{\boldsymbol{\Phi}}[p(\boldsymbol{W}|\boldsymbol{D})]$ of $\boldsymbol{W}$ given $\boldsymbol{D}$ in the approximated variational inference, $L_D(\boldsymbol{\Phi})$ in the RHS is the expected log-likelihood of DNN training, and $\mathrm{D_{KL}}(q_{\boldsymbol{\Phi}}(\boldsymbol{W}|\boldsymbol{Z})||p(\boldsymbol{W}))$ in the RHS is the KL divergence term from the approximated distribution of $\boldsymbol{W}$ to the prior distribution of $\boldsymbol{W}$ as a regularization term.

In the optimization of the approximated variational inference, we commonly maximize the RHS of (21), rather than maximizing the lower bound $L(\boldsymbol{\Phi})$ directly. Here, in SGVB inference, we approximate the expected log-likelihood $L_D(\boldsymbol{\Phi})$ by a mini-batch-form log-likelihood as we usually apply the mini-batch SGD algorithm for DNN training. The approximated log-likelihood $L_D^{\mathrm{SGVB}}(\boldsymbol{\Phi})$ for a mini batch can be considered as

$$\begin{aligned} L_D^{\mathrm{SGVB}}(\boldsymbol{\Phi}) &= \sum_{(\boldsymbol{x}_i, \boldsymbol{y}_i) \in (\boldsymbol{X}_b, \boldsymbol{Y}_b)} \int q_{\boldsymbol{\Phi}}(\boldsymbol{W}|\boldsymbol{Z}) \log p(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{W}) d\boldsymbol{W} \\ &= \frac{N}{N_b} \sum_{i=1}^{N_b} \log p(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{W} = f(\boldsymbol{\epsilon}; \boldsymbol{\Phi}, \boldsymbol{Z})) \\ &\approx L_D(\boldsymbol{\Phi}), \end{aligned} \tag{22}$$

where $(\boldsymbol{X}_b, \boldsymbol{Y}_b) = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N_b}$ is the bth mini batch stochastically selected from $\boldsymbol{D}$, $N_b = |(\boldsymbol{X}_b, \boldsymbol{Y}_b)|$ is the mini batch size, and $f(\boldsymbol{\epsilon}; \boldsymbol{\Phi}, \boldsymbol{Z})$ is a differentiable function for reparameterizing the dropout mask in the set $\boldsymbol{M} = \{\boldsymbol{m}^{(l)}\}_{l=1}^{L}$ by random samples $\boldsymbol{\epsilon}$, and for generating $\boldsymbol{W}$. Parameters in $\boldsymbol{\Lambda}$ are estimated directly by (22), respectively.

For the lth layer as an example, the dropout-masked parameter matrix $\boldsymbol{w}^{(l)}$ can be computed as

$$\begin{aligned} \boldsymbol{w}^{(l)} &= f\left(\boldsymbol{\epsilon}^{(l)}; \boldsymbol{\theta}^{(l)}, \boldsymbol{\lambda}_l\right) \\ &= \mathrm{diag}(\boldsymbol{m}^{(l)})\boldsymbol{\theta}^{(l)} \\ &= \mathrm{diag}\left(\mathrm{Sigmoid}(\mu_l + \sigma_l \times \boldsymbol{\epsilon}^{(l)})\right)\boldsymbol{\theta}^{(l)}, \end{aligned} \tag{23}$$

where element $\epsilon_j^{(l)} \sim \mathcal{N}(0, 1), j = 1, \cdots, K_l$ in $\boldsymbol{\epsilon}^{(l)}$ is randomly sampled from the normal distribution with zero mean and unit variance.

5 Experimental Results And Discussions

5.1 Advanced Dropout for Classification

5.1.1 Datasets

We evaluated the advanced dropout technique on seven image classification datasets, including MNIST [43], CIFAR-10 and -100 [44], *mini*ImageNet [45], Caltech-256 [46], ImageNet-32×32 [47], and ImageNet [48] datasets. Note that in the *mini*ImageNet dataset, 500 and 100 images per class were randomly selected from the training set of the full ImageNet dataset as the training set and the test set of this work, respectively. In the Caltech-256 dataset, 60 samples were randomly selected from each class, gathering as the training set. The remaining samples were for the test set. The ImageNet-32×32 dataset is more difficult than the full ImageNet dataset (with 224×224 images in general), since all the images are downsampled to 32×32 for both training and test [47]. The former five ones are small-scale datasets, while the ImageNet-32×32 and the ImageNet datasets are large-scale ones. We will separately discuss the two kinds.

5.1.2 Implementation Details

For the MNIST dataset, a FC neural network with two hidden layers and 800 hidden nodes each was constructed for the evaluation, while VGG16 [1] and ResNet18 [2] models were considered as the base models for all the other datasets.

In DNN training on the MNIST dataset, we trained the FC neural network with different dropout techniques for 200 epochs with the fixed learning rate as 0.01. For the ImageNet-32×32 dataset, we trained the models under 100 epochs with the initialized learning rate as 0.1 and decayed by a factor of 10 at the 50th and the 75th epochs, respectively. For all the other datasets, we trained the models 300 epochs with the initialized learning rate as 0.1 and decayed by a factor of 10 at the 150th and the 225th epochs, respectively. The batch size on each dataset was set as 256. In addition, we adopted the SGD optimizer, of which the momentum and the weight decay values were set as 0.9 and 5×10^{-4}, respectively. All the models with the advanced dropout technique and the referred techniques have been experimented 5 runs with random initialization, while we compared the results of the advanced dropout technique on the ImageNet dataset (with standard experimental settings) with those reported in references [18], [34]. The means and the standard deviations of the classification accuracies are presented in the following sections for comparison.

Meanwhile, nine referred techniques were selected based on their distributions of the dropout masks including standard dropout with Bernoulli noise ("dropout, Bernoulli") [14], Gaussian noise ("dropout, Gaussian") [32], and uniform noise ("dropout, uniform") [18], variational dropout [19], concrete dropout [33], continuous dropout [18], β-dropout [34], information dropout [16], and soft dropout [15]. We reimplemented and compared them with the advanced dropout technique. The two adaptive version of the soft dropout, *i.e.*, Gaussian soft dropout and Laplace soft dropout, were implemented on all the datasets for comparison, respectively. For the standard dropout with Bernoulli noise and Gaussian noise, we fixed the dropout rate as 0.5 during training. For the continuous dropout, we selected the variance σ^2 of its Gaussian prior from the set

TABLE 1: Test accuracies (%) on MNIST, CIFAR-10, CIFAR-100, *mini*ImageNet, and Caltech-256 datasets. Note that the best results are marked in **bold** and the second best results are marked by underline, respectively.

Dataset Base model	MNIST 784-2×800-10	CIFAR-10 VGG16/ResNet18	CIFAR-100 VGG16/ResNet18	*mini*ImageNet VGG16/ResNet18	Caltech-256 VGG16/ResNet18
No dropout	98.23 ± 0.11	$93.86 \pm 0.10/94.83 \pm 0.08$	$73.62 \pm 0.38/76.44 \pm 0.14$	$76.35 \pm 0.08/71.80 \pm 0.02$	$63.87 \pm 0.07/61.36 \pm 0.10$
Dropout, Bernoulli	98.46 ± 0.06	$93.81 \pm 0.14/94.84 \pm 0.07$	$73.77 \pm 0.14/76.67 \pm 0.22$	$76.21 \pm 0.09/72.02 \pm 0.40$	$64.76 \pm 0.14/61.64 \pm 0.10$
Dropout, Gaussian	98.45 ± 0.05	$93.83 \pm 0.14/95.02 \pm 0.14$	$73.78 \pm 0.19/76.62 \pm 0.14$	$76.14 \pm 0.09/71.98 \pm 0.45$	$64.47 \pm 0.22/61.97 \pm 0.29$
Dropout, uniform	98.50 ± 0.12	$93.82 \pm 0.09/94.86 \pm 0.14$	$73.76 \pm 0.17/76.76 \pm 0.11$	$76.84 \pm 0.07/72.07 \pm 0.02$	$64.43 \pm 0.10/\underline{62.14 \pm 0.08}$
Concrete dropout	98.45 ± 0.04	$93.79 \pm 0.14/94.99 \pm 0.11$	$73.67 \pm 0.16/76.48 \pm 0.26$	$76.35 \pm 0.08/71.56 \pm 0.11$	$63.79 \pm 0.07/62.12 \pm 0.08$
Variational dropout	98.46 ± 0.14	$93.81 \pm 0.10/95.05 \pm 0.10$	$73.98 \pm 0.25/76.76 \pm 0.38$	$76.56 \pm 0.07/72.06 \pm 0.04$	$64.46 \pm 0.11/61.98 \pm 0.18$
β-dropout	98.62 ± 0.09	$93.95 \pm 0.19/95.07 \pm 0.11$	$74.03 \pm 0.10/76.79 \pm 0.28$	$\underline{77.13 \pm 0.13}/72.24 \pm 0.06$	$\underline{64.84 \pm 0.07}/62.14 \pm 0.08$
Continuous dropout	98.45 ± 0.20	$93.86 \pm 0.08/94.92 \pm 0.10$	$73.85 \pm 0.21/76.90 \pm 0.27$	$\underline{76.71 \pm 0.13}/\underline{72.33 \pm 0.04}$	$64.69 \pm 0.09/61.80 \pm 0.16$
Information dropout	98.22 ± 0.25	$93.88 \pm 0.18/94.97 \pm 0.16$	$73.70 \pm 0.49/76.47 \pm 0.28$	$76.44 \pm 0.08/\underline{71.90 \pm 0.06}$	$64.11 \pm 0.13/61.76 \pm 0.09$
Gaussian soft dropout	98.64 ± 0.04	$\underline{93.97 \pm 0.24}/\underline{95.09 \pm 0.09}$	$\underline{74.07 \pm 0.38}/\underline{77.22 \pm 0.23}$	$76.56 \pm 0.05/71.74 \pm 0.03$	$63.80 \pm 0.10/58.68 \pm 0.24$
Laplace soft dropout	$\underline{98.70 \pm 0.10}$	$93.95 \pm 0.11/95.03 \pm 0.11$	$74.05 \pm 0.23/77.13 \pm 0.27$	$76.61 \pm 0.06/71.55 \pm 0.02$	$64.60 \pm 0.05/57.22 \pm 1.15$
Advanced dropout	$\mathbf{98.89 \pm 0.04}$	$\mathbf{94.28 \pm 0.03/95.52 \pm 0.09}$	$\mathbf{74.94 \pm 0.24/77.78 \pm 0.08}$	$\mathbf{77.35 \pm 0.01/72.89 \pm 0.06}$	$\mathbf{65.09 \pm 0.03/62.53 \pm 0.01}$

TABLE 2: The p-values of student's t-tests between the accuracies of the advanced dropout technique and all the referred techniques with different base models on MNIST, CIFAR-10, CIFAR-100, *mini*ImageNet, and Caltech-256 datasets, respectively. The significance level was 0.05. The advanced dropout technique has statistically significant difference from a referred technique if the corresponding p-value is smaller than 0.05.

Dataset Base model	MNIST 784-2×800-10	CIFAR-10 VGG16/ResNet18	CIFAR-100 VGG16/ResNet18	*Mini*ImageNet VGG16/ResNet18	Caltech-256 VGG16/ResNet18
No dropout	9.04×10^{-5}	$2.92 \times 10^{-4}/1.31 \times 10^{-6}$	$3.59 \times 10^{-4}/1.69 \times 10^{-6}$	$6.16 \times 10^{-6}/2.81 \times 10^{-8}$	$1.09 \times 10^{-7}/8.60 \times 10^{-6}$
Dropout, Bernoulli	7.32×10^{-6}	$1.06 \times 10^{-3}/8.69 \times 10^{-7}$	$1.92 \times 10^{-5}/1.30 \times 10^{-4}$	$7.44 \times 10^{-6}/5.62 \times 10^{-3}$	$8.43 \times 10^{-3}/2.64 \times 10^{-5}$
Dropout, Gaussian	1.23×10^{-6}	$1.37 \times 10^{-3}/3.39 \times 10^{-4}$	$1.93 \times 10^{-5}/2.56 \times 10^{-6}$	$4.26 \times 10^{-6}/4.84 \times 10^{-3}$	$1.91 \times 10^{-3}/6.67 \times 10^{-3}$
Dropout, uniform	2.25×10^{-3}	$1.08 \times 10^{-4}/6.39 \times 10^{-5}$	$1.55 \times 10^{-5}/7.18 \times 10^{-7}$	$6.62 \times 10^{-5}/2.94 \times 10^{-7}$	$6.53 \times 10^{-5}/3.34 \times 10^{-4}$
Concrete dropout	3.00×10^{-7}	$8.43 \times 10^{-4}/5.18 \times 10^{-5}$	$9.07 \times 10^{-6}/1.62 \times 10^{-4}$	$6.07 \times 10^{-6}/6.74 \times 10^{-7}$	$1.35 \times 10^{-7}/2.45 \times 10^{-4}$
Variational dropout	2.32×10^{-3}	$1.86 \times 10^{-4}/7.19 \times 10^{-5}$	$2.14 \times 10^{-4}/2.60 \times 10^{-3}$	$1.12 \times 10^{-5}/7.26 \times 10^{-9}$	$1.24 \times 10^{-4}/1.76 \times 10^{-3}$
β-dropout	4.10×10^{-3}	$1.18 \times 10^{-2}/1.25 \times 10^{-4}$	$1.51 \times 10^{-4}/7.66 \times 10^{-4}$	$1.30 \times 10^{-2}/1.76 \times 10^{-7}$	$1.77 \times 10^{-3}/3.34 \times 10^{-4}$
Continuous dropout	9.67×10^{-3}	$1.12 \times 10^{-4}/8.41 \times 10^{-6}$	$4.11 \times 10^{-5}/1.03 \times 10^{-3}$	$2.76 \times 10^{-4}/1.24 \times 10^{-7}$	$5.47 \times 10^{-4}/4.30 \times 10^{-4}$
Information dropout	2.73×10^{-3}	$3.76 \times 10^{-3}/2.52 \times 10^{-4}$	$1.24 \times 10^{-3}/1.16 \times 10^{-4}$	$8.20 \times 10^{-6}/2.22 \times 10^{-9}$	$2.28 \times 10^{-5}/2.23 \times 10^{-5}$
Gaussian soft dropout	5.58×10^{-5}	$1.21 \times 10^{-2}/4.35 \times 10^{-5}$	$1.88 \times 10^{-3}/7.72 \times 10^{-3}$	$1.28 \times 10^{-6}/3.12 \times 10^{-8}$	$1.19 \times 10^{-6}/1.73 \times 10^{-6}$
Laplace soft dropout	2.91×10^{-2}	$9.80 \times 10^{-4}/3.46 \times 10^{-5}$	$1.53 \times 10^{-4}/2.05 \times 10^{-3}$	$4.59 \times 10^{-6}/1.09 \times 10^{-7}$	$1.52 \times 10^{-7}/2.47 \times 10^{-4}$

$\{0.2, 0.3, 0.4\}$, following the strategy in [18]. Meanwhile, for the β-dropout, the shape parameter β was set as values drawn from the set $\{0.001, 0.1, 0.2, 0.5, 1, 3\}$, as suggested in [34]. For the information dropout, the multiplier β in its loss function was empirically selected as 0.1.

In order to check whether the advanced dropout technique has statistically significant performance improvement compared with the referred techniques, we conducted Student's t-tests between accuracies of them with the null-hypothesis that the means of two populations are equal. The significance level was set as 0.05.

5.1.3 Performance on MNIST Dataset

From Table 1, the advanced dropout technique achieves the best accuracy at 98.89% among the other techniques with the two-hidden-layer FC neural network on the MNIST dataset. At the meantime, it obtains the smallest standard deviation at 0.04% as well. While the second best technique, the Laplace soft dropout, achieves the classification accuracy at 98.70%, the proposed technique outperforms the second best one slightly at about 0.2%, but has an increase at about 0.7% compared with the model without dropout.

5.1.4 Performance on CIFAR-10 and -100 Datasets

On the CIFAR-10 dataset, the advanced dropout technique performs best again with both VGG16 and ResNet18 models. It achieves the averaged classification accuracies of 94.28% and 95.52% for each base model, respectively, while the accuracies of all the others are less than 94% and 95.1% with the VGG16 and the ResNet18 base models, respectively. Meanwhile, the proposed technique outperforms the second best technique, by about 0.3% and 0.5%, and the base model by about 0.4% and 0.7%, respectively.

Moreover, the advanced dropout technique achieves the classification accuracies of 74.94% and 77.78% on the CIFAR-100 dataset with the VGG16 and the ResNet18 base models, respectively, which are the best results among all the referred techniques. The VGG16 model with the advanced dropout technique obtains an improvement of 0.9% and 1.3%, compared with the Gaussian soft dropout and the base model, while the ResNet18 with it also achieves more than 0.5% and 1.3% higher performance compared with the Gaussian soft dropout and the base model.

5.1.5 Performance on MiniImageNet Dataset

In Table 1, the advanced dropout technique achieves the classification accuracies at 77.35% and 72.89%, respectively, which is the best results with the VGG16 and the ResNet18 models among the referred methods on the *mini*ImageNet dataset. The accuracies obtained by most of the referred techniques with the VGG16 model are lower than 77%. The proposed technique outperforms the base model by 1% and the second best model, the β-dropout with the VGG16 model, by more than 0.2%.

Furthermore, the advanced dropout-based ResNet18 model also shows improvement in classification accuracies on the *mini*ImageNet dataset. Compared with the base model, the advanced dropout technique with the ResNet18 model increases its accuracy by about 1.1%, which is a promising performance improvement. Meanwhile, it surpasses continuous dropout with the ResNet18 (72.33%) by more than 0.5%.

5.1.6 Performance on Caltech-256 Dataset

In the last column of Table 1, the advanced dropout technique with the VGG16 and the ResNet18 models performs the best on the Caltech-256 dataset and achieve the averaged accuracies of 65.09% and 62.53% with the smallest standard deviations of each at 0.03% and 0.01%, respectively. Compared with their corresponding base models, the models based on the advanced dropout technique improve the accuracies by a large margin of around 1.2% for both. Applying the VGG16 model as the base model, the classification accuracy of the advanced dropout is higher than the second best model, the β-dropout, more than 0.2%. Meanwhile, with the ResNet18 base model, the advanced dropout technique shows considerable performance improvements compared with the base model and the second best model as well. It improves the accuracy by 0.4% more than the second best technique, the dropout with uniform noise.

TABLE 3: Test accuracies (%) and the p-values of Student's t-tests between the accuracies of the advanced dropout technique and all the referred techniques on the ImageNet-32×32 dataset. Note that the best results are marked in **bold** and the second best results are marked by underline, respectively. The significance level was set as 0.05.

Dataset	VGG16		ResNet18	
Base model	Top-1 acc./p-value	Top-5 acc./p-value	Top-1 acc./p-value	Top-5 acc./p-value
No dropout	$40.58 \pm 0.15/3.57 \times 10^{-8}$	$64.50 \pm 0.22/1.56 \times 10^{-6}$	$45.46 \pm 0.39/5.44 \times 10^{-4}$	$70.47 \pm 0.38/9.05 \times 10^{-4}$
Dropout, Bernoulli	$41.26 \pm 0.21/1.67 \times 10^{-5}$	$64.14 \pm 0.12/1.63 \times 10^{-8}$	$45.72 \pm 0.13/1.46 \times 10^{-6}$	$70.73 \pm 0.08/4.24 \times 10^{-7}$
Dropout, Gaussian	$41.33 \pm 0.14/3.36 \times 10^{-7}$	$64.22 \pm 0.08/9.44 \times 10^{-8}$	$45.00 \pm 0.07/7.09 \times 10^{-1}$	$69.74 \pm 0.19/6.72 \times 10^{-7}$
Dropout, uniform	$41.36 \pm 0.16/2.35 \times 10^{-6}$	$64.88 \pm 0.15/5.63 \times 10^{-7}$	$45.26 \pm 0.08/5.27 \times 10^{-0}$	$69.62 \pm 0.07/2.47 \times 10^{-9}$
Concrete dropout	$41.37 \pm 0.05/1.73 \times 10^{-7}$	$64.33 \pm 0.19/2.90 \times 10^{-7}$	$45.75 \pm 0.12/7.26 \times 10^{-7}$	$70.55 \pm 0.17/5.68 \times 10^{-6}$
Variational dropout	$41.45 \pm 0.06/2.29 \times 10^{-7}$	$64.35 \pm 0.10/5.48 \times 10^{-8}$	$45.53 \pm 0.09/4.26 \times 10^{-9}$	$69.63 \pm 0.14/1.09 \times 10^{-8}$
β-dropout	$41.51 \pm 0.08/8.15 \times 10^{-8}$	$64.27 \pm 0.13/2.85 \times 10^{-8}$	$45.09 \pm 0.08/1.47 \times 10^{-10}$	$69.73 \pm 0.13/6.11 \times 10^{-9}$
Continuous dropout	$41.23 \pm 0.11/2.63 \times 10^{-8}$	$64.07 \pm 0.08/8.76 \times 10^{-8}$	$44.60 \pm 0.12/6.45 \times 10^{-9}$	$69.39 \pm 0.09/5.56 \times 10^{-10}$
Information dropout	$41.35 \pm 0.12/1.50 \times 10^{-7}$	$64.12 \pm 0.09/6.00 \times 10^{-8}$	$45.55 \pm 0.07/6.71 \times 10^{-10}$	$70.31 \pm 0.12/7.02 \times 10^{-8}$
Gaussian soft dropout	$41.69 \pm 0.07/4.30 \times 10^{-7}$	$64.59 \pm 0.10/2.47 \times 10^{-7}$	$46.33 \pm 0.13/4.06 \times 10^{-5}$	$71.04 \pm 0.06/1.09 \times 10^{-5}$
Laplace soft dropout	$41.57 \pm 0.14/1.33 \times 10^{-6}$	$65.45 \pm 0.10/2.88 \times 10^{-5}$	$46.31 \pm 0.19/3.85 \times 10^{-4}$	$71.14 \pm 0.06/3.14 \times 10^{-5}$
Advanced dropout	$\mathbf{42.65 \pm 0.11}$/ N/A	$\mathbf{66.23 \pm 0.16}$/ N/A	$\mathbf{47.03 \pm 0.07}$/ N/A	$\mathbf{71.75 \pm 0.12}$/ N/A

TABLE 4: Test accuracies (%) and the p-values of Student's t-tests between the accuracies of the advanced dropout technique and all the referred techniques on the ImageNet dataset. Note that the best results are marked in **bold** and the second best results are marked by underline, respectively. The significance level was set as 0.05. Note that "†" means the results in the row are reported in [18] and "‡" means the results in the row are reported in [34].

Method	Top-1 acc./p-value	Top-5 acc./p-value
No dropout	$73.42 \pm 0.059/2.43 \times 10^{-11}$	$91.68 \pm 0.025/6.30 \times 10^{-10}$
Adaptive dropout†	$73.73 \pm 0.046/2.65 \times 10^{-16}$	$91.59 \pm 0.061/6.56 \times 10^{-12}$
DropConnect†	$73.18 \pm 0.050/3.70 \times 10^{-18}$	$91.44 \pm 0.037/2.04 \times 10^{-15}$
Dropout, Bernoulli†	$73.01 \pm 0.065/2.71 \times 10^{-17}$	$91.14 \pm 0.042/1.36 \times 10^{-16}$
Dropout, Gaussian†	$74.21 \pm 0.045/3.60 \times 10^{-13}$	$92.01 \pm 0.065/1.98 \times 10^{-6}$
Dropout, uniform†	$74.09 \pm 0.046/4.67 \times 10^{-14}$	$91.92 \pm 0.048/2.25 \times 10^{-9}$
β-dropout‡	$74.25 \pm 0.052/4.92 \times 10^{-12}$	$92.04 \pm 0.046/2.64 \times 10^{-7}$
Continuous dropout†	$74.21 \pm 0.045/3.60 \times 10^{-13}$	$92.01 \pm 0.065/1.98 \times 10^{-6}$
Advanced dropout	$\mathbf{74.82 \pm 0.017}$/N/A	$\mathbf{92.26 \pm 0.029}$/N/A

5.1.7 Performance on ImageNet-32×32 Dataset

In this section, we evaluate the proposed advanced dropout on a large-scale dataset, *i.e.*, the ImageNet-32×32 dataset. The accuracies and p-values are listed in Table 3. According to the table, the advanced dropout achieves statistically significant improvements over all the referred methods, improving top-1 accuracy by about 1% and 0.7% and top-5 accuracy by about 0.8% and 0.6% with VGG16 and ResNet18 models, respectively.

5.1.8 Performance on ImageNet Dataset

In this section, the proposed advanced dropout was evaluated on the ImageNet dataset with VGG16 model as the base model. The accuracies and the corresponding p-values are listed in Table 4. According to the table, the advanced dropout achieves statistically significant improvements over all the referred methods, improving top-1 accuracy by about 0.6% and top-5 accuracy by about 0.2%, respectively.

5.1.9 Trade-off between Performance and Time

In this section, we compare the training time (in second/epoch) of the proposed advanced dropout with those of the other dropout variants by using one NVIDIA GTX1080Ti. Note that the dropout techniques for regularization are only used when training the models. Their inference time is the same as the models without dropouts. Thus, we only compare the training time on MNIST, CIFAR-10, CIFAR-100, *Mini*ImageNet, Caltech-256, and ImageNet-32×32 datasets, respectively. According to the results shown in Table 5, the models without dropouts always take the shortest time. Meanwhile, the proposed advanced dropout indeed required relatively longer time in each epoch, but it is not the slowest one. Moreover, the differences among the dropout techniques are not large. This indicates that time consumption for the dropout technique is mainly for sampling the dropout masks and the parametric prior does not take much training effort.

Furthermore, in order to quantitatively compare the accuracy improvement over the training time, we design a effectiveness ratio, which is similar to the cost effectiveness in [49], to rank all the dropout techniques based on their differences from the base model without dropout. A higher effectiveness ratio means more accuracy improvement can be achieved over the same unit of training time cost. We define the two metrics for the calculation of the effectiveness ratio and name them as "metric 1" and "metric 2", respectively. For the "metric 1", the effectiveness ratio s_1 is

$$s_1 = \frac{\text{Sigmoid}\,(a - a')}{\text{Sigmoid}\,((t - t')/t')}, \tag{24}$$

where a and t are the accuracy (in $\%$) and training time (in second/epoch) of a dropout technique, and a' and t' are those of the base model with no dropout. The Sigmoid functions here are for scaling the accuracy improvement and the training time into the same range. For the "metric 2", the Sigmoid functions are removed and the effectiveness ratio s_2 is

$$s_2 = \frac{(a - a')/a'}{(t - t')/t'}. \tag{25}$$

For better illustration, we normalize all the effectiveness ratios of different dropout techniques from one dataset by their maximum and minimum values. The normalized effectiveness ratio s^i_{norm} of the "metric i" ($i = 1, 2$) for one specific dataset is defined as

$$s^i_{\text{norm}} = \frac{s - \min_{s'\in\mathcal{S}_i} s'}{\max_{s'\in\mathcal{S}_i} s' - \min_{s'\in\mathcal{S}_i} s'}, s \in \mathcal{S}_i, \tag{26}$$

where $\mathcal{S}_i, i = 1, 2$ is the set of effectiveness ratios of the dropout techniques calculated by the two metrics, respectively. Here, "normalized" means mapping the effectiveness ratios from one dataset and in a base model (called a case) into the interval of $[0, 1]$ for comparing their relative values.

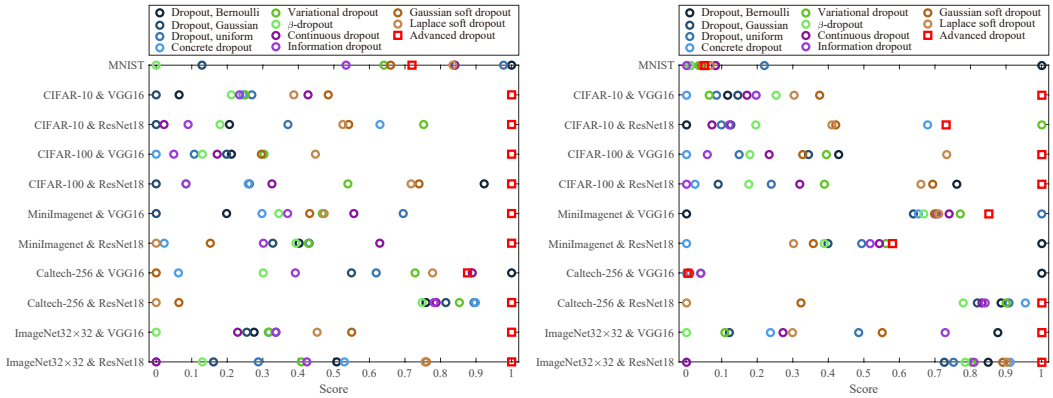
Figure 5 demonstrates the normalized effectiveness ratios of different dropout techniques on each dataset. By using "metric 1" (Figure 5(a)), the advanced dropout can outperform all the other dropout techniques on nine cases out of all the eleven cases. Meanwhile, with "metric 2" (Figure 5(b)), the advanced dropout can outperform all the other dropout techniques on six cases in all eleven cases and perform the second best on three cases.

5.1.10 Performance with other Base Models

We further evaluated the advanced dropout technique with three other base models, including DenseNet40 [3], MobileNet [50], wide residual network (WRN) [51], on both the

TABLE 5: Comparison of training time per epoch (in second/epoch) on MNIST, CIFAR-10, CIFAR-100, *mini*ImageNet, Caltech-256, and ImageNet-32 × 32 datasets by using one NVIDIA GTX1080Ti. All the models with different dropout variants are trained under equal conditions. The shortest time and the longest time on each dataset are highlighted in **bold** and *italic*, respectively.

Dataset Base model	MNIST 784-2 × 800-10	CIFAR-10 VGG16/ResNet18	CIFAR-100 VGG16/ResNet18	*Mini*ImageNet VGG16/ResNet18	Caltech-256 VGG16/ResNet18	ImageNet-32 × 32 VGG16/ResNet18
No dropout	**3.95**	**18.34/38.07**	**45.41/20.62**	**301.55/174.75**	**127.74/60.20**	**406.87/556.13**
Dropout, Bernoulli	4.00	23.32/44.65	48.81/26.95	302.47/175.18	127.77/62.50	411.62/572.08
Dropout, Gaussian	9.45	*24.69/57.83*	49.77/53.54	365.86/183.41	148.12/70.09	438.49/579.57
Dropout, uniform	4.22	20.87/41.87	52.55/46.56	307.60/177.50	136.59/65.71	416.47/572.70
Concrete dropout	4.55	20.57/41.58	52.93/36.79	308.47/175.54	128.62/64.35	425.68/564.60
Variational dropout	5.47	20.89/41.38	54.17/37.28	309.19/176.44	128.27/64.81	446.90/581.87
β-dropout	*15.15*	24.27/55.00	*63.84/57.75*	*520.72/210.97*	*221.21/90.25*	*605.93/696.83*
Continuous dropout	4.55	19.81/52.66	53.93/49.33	319.40/178.57	132.48/66.20	420.53/559.86
Information dropout	4.93	22.52/53.06	52.21/42.44	309.12/175.60	127.96/65.11	413.30/575.36
Gaussian soft dropout	6.15	21.25/47.10	58.17/44.10	321.40/179.45	138.53/67.18	418.97/586.96
Laplace soft dropout	5.52	22.01/45.19	51.49/42.39	321.01/179.04	135.81/66.35	425.97/582.75
Advanced dropout	6.87	21.08/52.16	59.43/48.93	323.18/181.20	163.28/65.41	419.59/582.22



(a) Normalized effectiveness ratio with "metric 1" (b) Normalized effectiveness ratio with "metric 2"

Fig. 5: Normalized effectiveness ratios of the advanced dropout technique and various dropout techniques for quantitative comparison of accuracy improvement over training time. We show the results with the two metrics in (a) and (b), respectively. The red squares are the proposed advanced dropout technique, while other circles represent the referred dropout techniques. Note that for the ImageNet-32 × 32 dataset, top-1 accuracies are used only.

CIFAR-10 and -100 datasets. For hyperparameter selection of the base models, the growth rate and the reduction rate were set as 12 and 0.5 for the DenseNet40, respectively. α was set as 1 for the MobileNet, and k was set as 10 for the WRN16. The optimizer, the batch sizes, and the learning rates were set the same as those of the previous settings. All the models with the advanced dropout technique and the referred techniques were experimented 5 runs with random initialization. In addition, Student's t-tests was conducted between the accuracies of the advanced dropout technique and all the referred techniques.

The accuracies and the p-values are listed in Table 6. It can be observed that the advanced dropout achieves statistically significant improvement on all the base models.

TABLE 6: Test accuracies (%) and p-values of Student's t-tests between the accuracies of the advanced dropout technique and all the referred techniques with other base models on CIFAR-10 and -100 datasets, respectively. Note that the best results are marked in **bold** and the second best results are marked by underline, respectively. The significance level was set as 0.05.

Dataset Base model	CIFAR-10 DenseNet40/p-value	 MobileNet/p-value	 WRN16-10/p-value
No dropout	$92.54 \pm 0.33/1.47 \times 10^{-3}$	$90.82 \pm 0.12/8.99 \times 10^{-6}$	$94.29 \pm 0.37/9.36 \times 10^{-4}$
Dropout, Bernoulli	$92.06 \pm 0.26/4.96 \times 10^{-5}$	$90.95 \pm 0.12/1.32 \times 10^{-5}$	$94.65 \pm 0.40/4.59 \times 10^{-3}$
Dropout, Gaussian	$92.05 \pm 0.25/3.70 \times 10^{-5}$	$91.23 \pm 0.19/2.04 \times 10^{-3}$	$94.93 \pm 0.12/2.22 \times 10^{-4}$
Dropout, uniform	$92.41 \pm 0.21/6.69 \times 10^{-5}$	$90.95 \pm 0.30/2.36 \times 10^{-3}$	$94.83 \pm 0.35/6.59 \times 10^{-3}$
Concrete dropout	$92.50 \pm 0.26/3.95 \times 10^{-4}$	$91.04 \pm 0.17/3.09 \times 10^{-4}$	$94.94 \pm 0.05/1.25 \times 10^{-6}$
Variational dropout	$92.71 \pm 0.15/2.87 \times 10^{-5}$	$91.17 \pm 0.27/5.71 \times 10^{-3}$	$\underline{94.98 \pm 0.11}/1.65 \times 10^{-4}$
β-dropout	$92.41 \pm 0.21/5.18 \times 10^{-5}$	$90.94 \pm 0.25/9.98 \times 10^{-4}$	$94.83 \pm 0.12/1.02 \times 10^{-4}$
Continuous dropout	$92.08 \pm 0.18/3.20 \times 10^{-6}$	$90.85 \pm 0.34/2.47 \times 10^{-3}$	$94.95 \pm 0.13/3.38 \times 10^{-4}$
Information dropout	$92.68 \pm 0.15/2.14 \times 10^{-5}$	$91.29 \pm 0.20/4.82 \times 10^{-3}$	$94.94 \pm 0.13/3.19 \times 10^{-4}$
Gaussian soft dropout	$\underline{92.73 \pm 0.17}/8.64 \times 10^{-5}$	$\underline{91.34 \pm 0.22}/1.17 \times 10^{-2}$	$94.97 \pm 0.07/1.94 \times 10^{-5}$
Laplace soft dropout	$92.71 \pm 0.20/2.20 \times 10^{-4}$	$91.34 \pm 0.25/1.65 \times 10^{-2}$	$94.87 \pm 0.05/9.74 \times 10^{-7}$
Advanced dropout	$\mathbf{93.41 \pm 0.09}$/ N/A	$\mathbf{91.76 \pm 0.05}$/ N/A	$\mathbf{95.50 \pm 0.02}$/ N/A
Dataset Base model	**CIFAR-100 DenseNet40/p-value**	** MobileNet/p-value**	** WRN16-10/p-value**
No dropout	$68.36 \pm 0.27/6.75 \times 10^{-7}$	$67.12 \pm 0.25/1.89 \times 10^{-6}$	$77.67 \pm 0.19/2.83 \times 10^{-3}$
Dropout, Bernoulli	$68.39 \pm 0.27/7.10 \times 10^{-7}$	$67.39 \pm 0.51/7.27 \times 10^{-4}$	$77.55 \pm 0.18/1.35 \times 10^{-4}$
Dropout, Gaussian	$68.43 \pm 0.31/3.50 \times 10^{-6}$	$67.30 \pm 0.41/1.25 \times 10^{-4}$	$77.68 \pm 0.16/4.95 \times 10^{-4}$
Dropout, uniform	$68.29 \pm 0.36/8.24 \times 10^{-6}$	$\underline{67.87 \pm 0.12}/8.34 \times 10^{-5}$	$77.63 \pm 0.14/3.80 \times 10^{-3}$
Concrete dropout	$68.48 \pm 0.49/1.19 \times 10^{-4}$	$67.87 \pm 0.23/7.79 \times 10^{-5}$	$77.69 \pm 0.16/5.24 \times 10^{-5}$
Variational dropout	$\underline{69.70 \pm 0.28}/2.38 \times 10^{-4}$	$67.81 \pm 0.27/1.04 \times 10^{-4}$	$77.73 \pm 0.17/1.26 \times 10^{-3}$
β-dropout	$68.52 \pm 0.35/1.08 \times 10^{-5}$	$67.26 \pm 0.36/4.02 \times 10^{-5}$	$77.78 \pm 0.12/1.09 \times 10^{-4}$
Continuous dropout	$68.43 \pm 0.33/5.09 \times 10^{-6}$	$67.12 \pm 0.36/2.62 \times 10^{-5}$	$77.56 \pm 0.14/1.28 \times 10^{-5}$
Information dropout	$69.40 \pm 0.25/1.69 \times 10^{-5}$	$67.69 \pm 0.36/2.68 \times 10^{-4}$	$\underline{77.85 \pm 0.05}/1.74 \times 10^{-4}$
Gaussian soft dropout	$69.20 \pm 0.16/3.61 \times 10^{-7}$	$67.75 \pm 0.32/1.83 \times 10^{-4}$	$77.83 \pm 0.11/6.03 \times 10^{-6}$
Laplace soft dropout	$69.46 \pm 0.13/1.46 \times 10^{-6}$	$67.8 \pm 0.13/4.93 \times 10^{-5}$	$77.74 \pm 0.19/2.94 \times 10^{-5}$
Advanced dropout	$\mathbf{70.65 \pm 0.17}$/ N/A	$\mathbf{68.85 \pm 0.24}$/ N/A	$\mathbf{78.22 \pm 0.15}$/ N/A

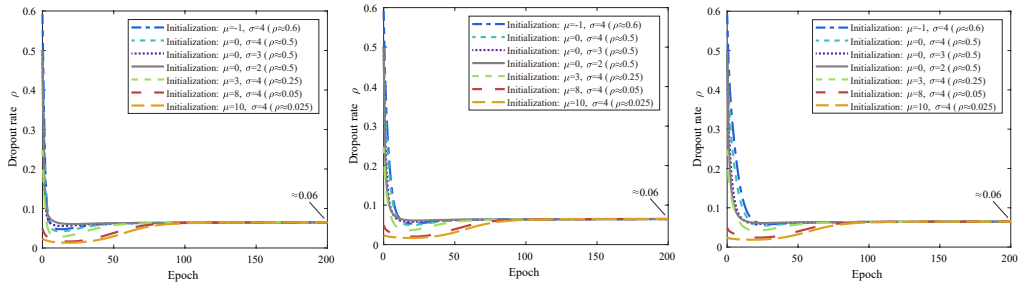
5.2 Ablation Studies

To investigate the effectiveness of the components of the advanced dropout technique, including the parametric prior and the SGVB inference, we conducted quantitative comparisons on the MNIST [43], the CIFAR-10 and -100 [44], the *mini*ImageNet [45], and the Caltech-256 [46] datasets. The experimental results are shown in Table 7. We compare the full advanced dropout technique with the techniques whose the parametric prior component is removed ("advanced dropout w/o prior" in Table 7) and whose parameters μ_l and σ_l are fixed at 0.5 ("advanced dropout w/o optimization" in Table 7). The model we applied is the two-hidden-layer FC neural network with 800 hidden nodes in each layer on the MNIST dataset, while the VGG16 model is used on the other datasets as the base models.

It can be observed that, the full advanced dropout technique outperforms the base model without any dropout training by about 0.7%, 0.5%, 1.3%, 1%, and 1.2% on the five datasets, respectively. When removing the parametric prior from the full technique, the classification accuracies decrease slightly on all the datasets, even though the technique performs better than the base model as well. When we continue removing the SGVB inference and setting the technique with fixed parameters μ_l and σ_l, classification accuracies further decrease. Therefore, the parametric prior and the SGVB inference play their own positive roles in the advanced dropout, which are both essential.

TABLE 7: Test accuracies (%) of ablation studies on MNIST, CIFAR-10, CIFAR-100, *mini*ImageNet, and Caltech-256 datasets. "Advanced dropout w/o optimization" means the advanced dropout with fixed parameters; "Advanced dropout w/o prior" mean the advanced dropout minus the prior. Note that the best results are marked in **bold**, respectively.

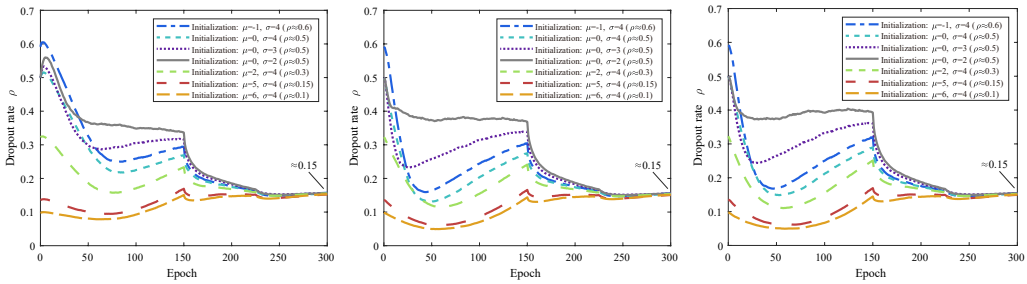
Dataset Base Model	MNIST 784-2×800-10	CIFAR-10 VGG16	CIFAR-100 VGG16	*Mini*ImageNet VGG16	Caltech-256 VGG16
No dropout	98.23 ± 0.11	93.86 ± 0.10	73.62 ± 0.38	76.35 ± 0.08	63.87 ± 0.07
Advanced dropout w/o optimization	98.70 ± 0.04	93.98 ± 0.09	74.55 ± 0.19	77.20 ± 0.08	64.88 ± 0.11
Advanced dropout w/o prior	98.81 ± 0.02	94.20 ± 0.07	74.77 ± 0.10	77.30 ± 0.08	65.03 ± 0.01
Advanced dropout	$\mathbf{98.89 \pm 0.04}$	$\mathbf{94.28 \pm 0.03}$	$\mathbf{74.94 \pm 0.24}$	$\mathbf{77.35 \pm 0.01}$	$\mathbf{65.09 \pm 0.03}$



Initialization	Test acc. (%)
$\mu = -1, \sigma = 4$ ($\rho \approx 0.6$)	98.85
$\mu = 0, \sigma = 4$ ($\rho \approx 0.5$)	98.89
$\mu = 0, \sigma = 3$ ($\rho \approx 0.5$)	98.94
$\mu = 0, \sigma = 2$ ($\rho \approx 0.5$)	98.87
$\mu = 3, \sigma = 4$ ($\rho \approx 0.25$)	98.91
$\mu = 8, \sigma = 4$ ($\rho \approx 0.05$)	98.90
$\mu = 10, \sigma = 4$ ($\rho \approx 0.025$)	98.90

(a) (b) (c) (d)

Fig. 6: Illustration of the adaptive dropout rates in the advanced dropout technique on the MNIST dataset. The subfigures illustrate the dropout rate curves of the advanced dropout of (a) the input features, (b) the first hidden layer, and (c) the second hidden layer. Note that differently initialized dropout rates ρ (see legends) are set for initialization of the parameters μ_l and σ_l. Please refer to (27) for the calculation formula of the dropout rate. Corresponding test accuracies (acc.) with different initializations are also reported in (d).



Initialization	Test acc. (%)
$\mu = -1, \sigma = 4$ ($\rho \approx 0.6$)	94.24
$\mu = 0, \sigma = 4$ ($\rho \approx 0.5$)	94.28
$\mu = 0, \sigma = 3$ ($\rho \approx 0.5$)	94.27
$\mu = 0, \sigma = 2$ ($\rho \approx 0.5$)	94.36
$\mu = 2, \sigma = 4$ ($\rho \approx 0.3$)	93.92
$\mu = 5, \sigma = 4$ ($\rho \approx 0.15$)	94.08
$\mu = 6, \sigma = 4$ ($\rho \approx 0.1$)	93.94

(a) (b) (c) (d)

Fig. 7: Illustration of the adaptive dropout rates in the advanced dropout technique on the CIFAR-10 dataset. The subfigures mean the dropout rate curves of the advanced dropout of (a) the convolutional features of the last convolutional layer, (b) the first hidden layer, and (c) the second hidden layer. Differently initialized dropout rates ρ (see legends) are set for initialization of the parameters μ_l and σ_l. Corresponding test accuracies (acc.) with different initializations are also reported in (d).

5.3 Analysis of Dropout Rate Characteristics

In this section, we extensively study the characteristics of the advanced dropout technique, especially its dropout rate learning process. Firstly, we analyze the convergence of the adaptive dropout rate and the learned distributions of the dropout masks. Then, to compare the advanced dropout technique with the one generating dropout rate directly (without an explicit distribution), we design a dropout variant in this kind and conduct several experiments.

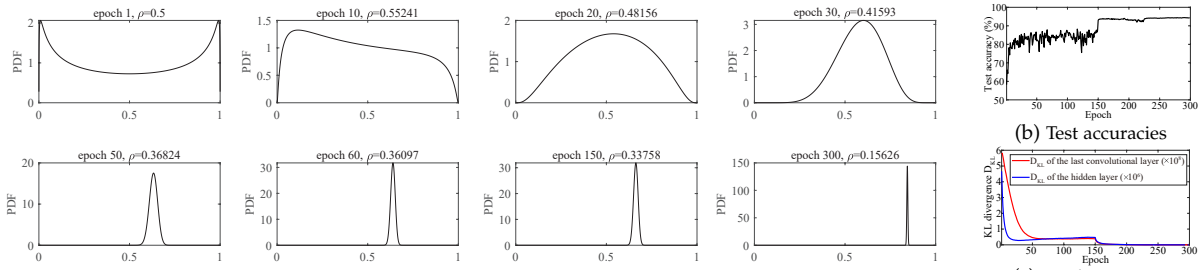
5.3.1 Convergence of Adaptive Dropout Rate

In this section, we discuss the convergence of the advanced dropout technique. Due to the importance of the distribution parameters μ_l and σ_l of each layer in DNNs, we investigate their convergence via a quantitative indicator, called dropout rate which is essential in dropout techniques. In standard dropout [14], the dropout rate is considered as one minus the parameter p of the Bernoulli distribution. Given that the parameter p represents the mean of the Bernoulli distribution, we can generalize the calculation of the dropout rate by one minus the mean of the distribution of the dropout masks. In particular, the dropout rate ρ_l of the advanced dropout technique for all the K_l nodes $m_1^{(l)}, \cdots, m_{K_l}^{(l)}$ can be explicitly calculated by μ_l and σ_l as

$$\rho_l = 1 - \mathrm{E}[m_j^{(l)}] \approx 1 - \mathrm{Sigmoid}\left(\frac{\mu_l}{\sqrt{1 + \frac{\pi}{8}\sigma_l^2}}\right). \tag{27}$$

In this case, we conducted groups of experiments and illustrate their dropout rate curves in Figure 6 and 7. The MNIST and the CIFAR-10 datasets were used for evaluations. For the MNIST dataset, a FC neural network with two hidden layers and 800 hidden nodes of each was trained for 200 epochs with the fixed learning rate of 0.01, while we applied a VGG16 model as the base model and trained it for 200 epochs on the CIFAR-10 dataset but with the dynamic learning rate same as that in Section 5.1.2.

In Figure 6 and 7, with different initial dropout rates, it can be observed that the dropout rate of the proposed advanced dropout technique converged to similar values on the two datasets, respectively (*i.e.*, ≈ 0.06 on the MNIST dataset and ≈ 0.15 on the CIFAR-10 dataset). This indicates that different initial dropout rates do not influence the optimization results and can converge to the similar last values. All the models with different initializations converge to the same values gradually after the 150th epoch. Therefore, we can conclude that the advanced dropout technique has good convergence performance. We should note that the initial dropout rate 0.5 (a commonly used value in [14], [18]) is an okay choice and the other values are also suitable.



(a) Learned distributions of different epochs (b) Test accuracies (c) KL divergences

Fig. 8: Learned distributions of the dropout masks with the VGG16 model on the CIFAR-10 dataset. Learned distributions of eight selected epochs are shown in (a). Note that for other datasets and base models, the distributions are similar with the shown ones. To further discuss the relationship between the convergence of the model and the distributions, we illustrate the curve of test accuracies in (b) and KL divergences from the distributions of different epochs to that of the last epoch in (c).

TABLE 8: Comparison of the advanced dropout and a dropout variant that directly computes the dropout rate by the MLP. The test accuracies (%) and the p-values of Student's t-tests on CIFAR-10 and -100 datasets are shown, respectively. Note that the best results are marked in **bold**.

Dataset	CIFAR-10		CIFAR-100	
Base model	VGG16/p-value	ResNet18/p-value	VGG16/p-value	ResNet18/p-value
Concrete+MLP	$94.10 \pm 0.08/3.87 \times 10^{-3}$	$95.08 \pm 0.04/4.13 \times 10^{-5}$	$73.77 \pm 0.19/1.76 \times 10^{-5}$	$76.45 \pm 0.19/1.77 \times 10^{-5}$
Advanced dropout	$\mathbf{94.28 \pm 0.03}$/ N/A	$\mathbf{95.52 \pm 0.09}$/ N/A	$\mathbf{74.94 \pm 0.24}$/ N/A	$\mathbf{77.78 \pm 0.08}$/ N/A

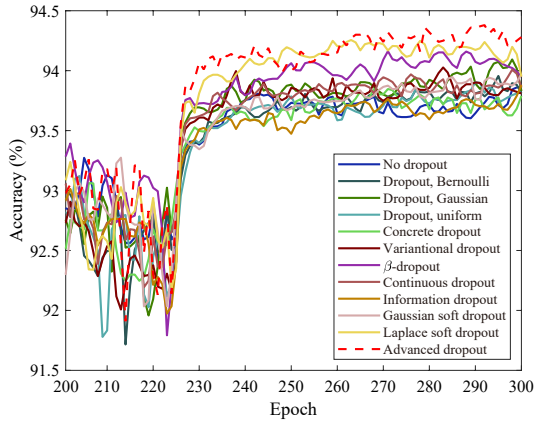


Fig. 9: Test accuracy curves with VGG16 on the CIFAR-10 dataset as an example. We only illustrate the last 100 epochs for better comparisons of the accuracies of different dropout variants and the proposed advanced dropout technique.

Meanwhile, the dropout rates for each layer are optimized and vary during training, which means that different epochs need different values of dropout rate. The low dropout rates at the last epochs are just suitable for the last epochs only.

An interesting phenomenon is observed in Figure 7. In the different layers, the dropout rates with distinct initializations fall immediately until reaching a local minimum of the dropout rate curves and then increase continuously. Before reaching the local minimum, the model fits the data rapidly and the dropout rates of different layers are adaptively reduced to a low level in a quick way for fully optimizing model parameters. Then after the local minimum, failing of convergence of the optimization (because of too large learning rate) occurs and a smaller learning rate is required to decrease optimization steps (*i.e.*, the learning rate multiplying gradients in SGD). As the decrease of the optimization steps can be also led to by multiplying smaller dropout masks onto the outputs of each layer (*i.e.*, increasing the dropout rates), the dropout rates are driven to rise automatically. In addition, at the 150th and the 225th epochs, the dropout rates fall off again due to the decay of the learning rate.

In addition, different initializations of μ and σ only slightly affect the test accuracies of the models, as demonstrated in Figure 6(d) and 7(d).

5.3.2 Learned Distributions of the Dropout Masks

We discuss the shapes of the learned distributions of the dropout masks, as shown in Figure 8(a). We only illustrate the distributions optimized with the VGG16 model on the CIFAR-10 dataset, since the distributions for other datasets and base models are similar with the shown ones. As shown in Figure 8(a), the distributions change dramatically between epoch 1 and epoch 60, and then converged till the end of the training process. Based on the view of Bayesian inference, the variance of an estimated posterior distribution will converge to a small value and the estimated posterior distribution will be concentrated in a compact area, if the data amount is large enough [42], [52].

To further discuss the relationship between the convergence of the model and the distributions, we illustrate the curve of test accuracies in Figure 8(b) and the curve of the KL divergences from the distributions on different epochs to that of the last epoch in Figure 8(c). The KL divergences decrease significantly at the beginning of training and then converge. During the whole training process, the KL divergences continuously decrease until the last epoch as the test accuracies increase. This means that the distributions are optimized gradually with model training.

5.3.3 Comparison with Dropout Rate Generation w/o Explicit Distribution

In this section, we compare the proposed advanced dropout (with an explicit distribution assumption) with the cases that compute the dropout rates directly (without explicit distribution assumption). To this end, we design a dropout variant with the concrete distribution [53], in which the parameter p, *i.e.*, one minus the dropout rate, is directly computed by an MLP. We name the designed dropout variant as "concrete+MLP". Table 8 lists the experimental results and the p-values of the Student's t-tests on the CIFAR-10 and -100 datasets, respectively. The advanced dropout achieves statistically significantly better performance than the "concrete+MLP" method on two datasets.

5.4 Capability of Overfitting Prevention

We illustrate the test accuracy curves of the last 100 epochs with VGG16 on the CIFAR-10 dataset as an example in

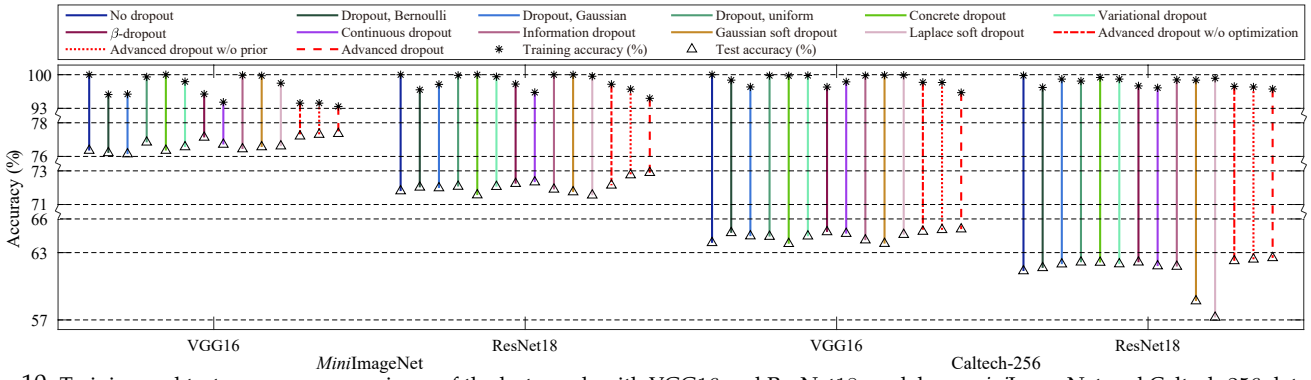


Fig. 10: Training and test accuracy comparisons of the last epoch with VGG16 and ResNet18 models on *mini*ImageNet and Caltech-256 datasets, respectively. The vertical lines in red are for the proposed advanced dropout, the advanced dropout w/o prior, and the advanced dropout w/o optimization, respectively. It can be observed that all the essential components in the proposed advanced dropout contribute to overfitting prevention. The other solid lines (with different colors) are for the referred methods.

TABLE 9: Top-1 and top-5 test accuracies (acc., %) with different training set sizes on Caltech-256 dataset. ξ means the training sample number per class. Note that the best results are marked in **bold**, respectively.

Model	$\xi = 60$		$\xi = 30$		$\xi = 15$	
	Top-1 acc.	Top-5 acc.	Top-1 acc.	Top-5 acc.	Top-1 acc.	Top-5 acc.
No dropout	63.87 ± 0.07	80.41 ± 0.07	51.12 ± 0.11	70.77 ± 0.08	36.19 ± 0.10	55.66 ± 0.15
Dropout, Bernoulli	64.76 ± 0.14	81.34 ± 0.14	51.88 ± 0.14	71.21 ± 0.14	36.31 ± 0.27	56.17 ± 0.24
Dropout, Gaussian	64.47 ± 0.22	81.05 ± 0.19	51.97 ± 0.14	71.36 ± 0.13	36.35 ± 0.10	56.21 ± 0.12
Dropout, uniform	64.43 ± 0.10	81.01 ± 0.10	51.78 ± 0.23	71.11 ± 0.11	35.82 ± 0.07	56.02 ± 0.06
Concrete dropout	63.79 ± 0.07	80.37 ± 0.07	50.72 ± 0.21	70.55 ± 0.11	35.08 ± 0.10	55.30 ± 0.09
Variational dropout	64.46 ± 0.11	81.04 ± 0.04	52.08 ± 0.08	70.91 ± 0.14	36.42 ± 0.14	56.14 ± 0.26
β-dropout	64.84 ± 0.07	81.07 ± 0.09	52.13 ± 0.13	71.33 ± 0.04	36.33 ± 0.04	56.28 ± 0.14
Continuous dropout	64.69 ± 0.09	81.06 ± 0.09	52.12 ± 0.18	71.31 ± 0.19	36.24 ± 0.22	56.16 ± 0.22
Information dropout	64.11 ± 0.13	81.55 ± 0.06	51.61 ± 0.11	70.92 ± 0.17	35.83 ± 0.13	55.97 ± 0.09
Gaussian soft dropout	63.80 ± 0.10	80.38 ± 0.09	51.01 ± 0.12	70.73 ± 0.14	34.65 ± 0.22	55.26 ± 0.15
Laplace soft dropout	64.60 ± 0.05	81.18 ± 0.04	51.61 ± 0.14	70.80 ± 0.06	35.49 ± 0.21	55.09 ± 0.18
Advanced dropout	$\mathbf{65.09 \pm 0.03}$	$\mathbf{81.64 \pm 0.04}$	$\mathbf{52.57 \pm 0.05}$	$\mathbf{71.62 \pm 0.13}$	$\mathbf{36.95 \pm 0.12}$	$\mathbf{56.58 \pm 0.16}$

TABLE 10: Test accuracies (%) of fully connected (FC) neural networks with different depth on MNIST dataset. The model structures mean "hidden layer number $\times$ hidden node number per layer". Note that the best results are marked in **bold**, respectively.

Model	FC (2×800)	FC (4×800)	FC (8×800)
No dropout	98.23 ± 0.11	98.16 ± 0.30	98.06 ± 0.47
Dropout, Bernoulli	98.46 ± 0.06	98.39 ± 0.10	98.18 ± 0.03
Dropout, Gaussian	98.45 ± 0.05	98.41 ± 0.15	98.05 ± 0.11
Dropout, uniform	98.50 ± 0.12	98.68 ± 0.06	98.40 ± 0.16
Concrete dropout	98.45 ± 0.04	98.54 ± 0.03	98.16 ± 0.21
Variational dropout	98.46 ± 0.14	98.34 ± 0.17	98.20 ± 0.12
β-dropout	98.62 ± 0.09	98.68 ± 0.06	98.46 ± 0.07
Continuous dropout	98.45 ± 0.20	98.42 ± 0.23	98.28 ± 0.07
Information dropout	98.22 ± 0.25	98.18 ± 0.22	97.65 ± 0.14
Gaussian soft dropout	98.64 ± 0.04	98.63 ± 0.05	98.67 ± 0.05
Laplace soft dropout	98.70 ± 0.10	98.53 ± 0.04	98.68 ± 0.06
Advanced dropout	$\mathbf{98.89 \pm 0.04}$	$\mathbf{98.99 \pm 0.04}$	$\mathbf{98.86 \pm 0.06}$

Figure 9. For the proposed advanced dropout technique, an increase trend can be clearly found and the curve varies slightly, although the best accuracy may be obtained in earlier epochs. Meanwhile, for some referred techniques, for example, the Laplace soft dropout, overfitting can be found between the 280th and 300th epochs. This means that the proposed advanced dropout technique can actually reduce overfitting. Similar results can be also found with other base models and on other datasets.

One way for overfitting prevention is to trade off the bias and the variance of a model to achieve small gap between training and test performance, *i.e.*, improved generalization. Here, we show the training and the corresponding test accuracies of the last epoch when applying different dropout techniques to the base models in Figure 10. It can be observed that the proposed advanced dropout technique reduces the gap between training and test accuracies (*i.e.*, reducing overfitting) and achieves the highest test accuracies at the same time. The models after removing the essential components (as discussed in Section 5.2) are still able to reduce the training-test accuracy gap compared with the referred methods, although they obtain suboptimal results than that with the "full" advanced dropout technique. This means all the essential components contribute to overfitting prevention.

To quantitatively investigate the capability of overfitting prevention of the advanced dropout technique, we designed two groups of experiments. Given the actual situation that overfitting occurs when the model size is too huge or the training dataset is too small, two groups of experiments are as follows: one exponentially decreases the training sample number in each class on the Caltech-256 dataset and the other exponentially increases the hidden layer number in a FC neural network on the MNIST dataset. The experimental results are shown in Table 9 and 10.

On the Caltech-256 dataset, a coefficient ξ is introduced to represent the training sample number per class and measure the size of the training set. In the experiments, ξ is set as $\{60, 30, 15\}$ with the test set fixed as the remaining part of the whole dataset by removing 60 samples from each class ($\xi = 60$). Moreover, we applied the VGG16 model as the base model for each case. The top-1 and the top-5 test accuracies are reported in Table 9. With the decrease of ξ, the classification accuracies of all the dropout techniques drop dramatically, which is expected. However, the advanced dropout technique achieves the best performance on the three cases with different ξ. When ξ is equal to 60, it obtains the top-1 and the top-5 accuracies of 65.09% and 81.64%, outperforming the second best technique by about 0.2% and 0.1%, respectively. At the mean time, it achieves the top-1 and top-5 accuracies of 52.57% and 71.62%, which

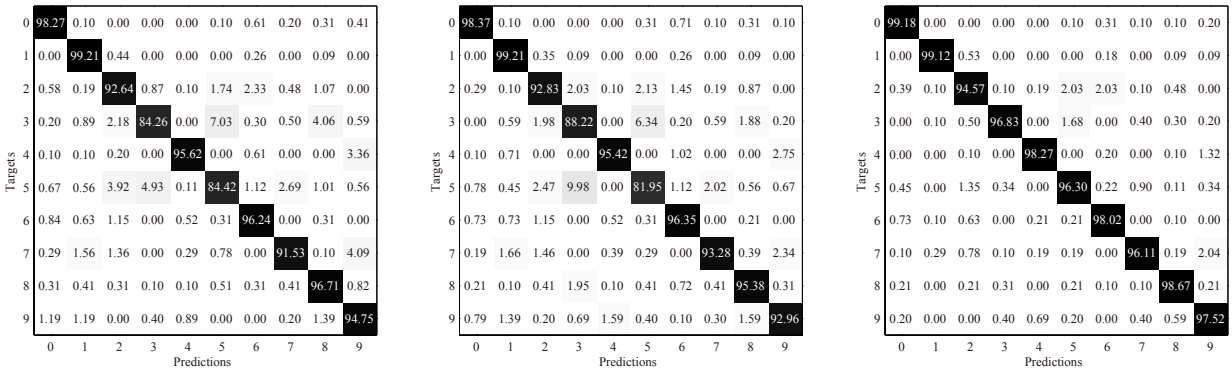


Fig. 11: Confusion matrices of different techniques for model uncertainty inference on MNIST dataset: (a) MC dropout [24], (b) ApDeepSense [54], and (c) advanced dropout.

TABLE 11: AUROC of max probability (Max.P) and entropy (Ent.), and test accuracies (acc., %) of model certainty inference on the MNIST dataset. Note that the best results are marked in **bold**.

Method	AUROC		Acc.
	Max.P	Ent.	
MC dropout	0.9210	0.8892	93.49
ApDeepSense	0.9185	0.8951	93.56
Advanced dropout	**0.9552**	**0.9357**	**97.47**

are 0.44% and 0.28% larger than the second best technique, respectively, with the training set reduced by half ($\xi = 30$). Furthermore, For $\xi = 15$, it surpasses the second best technique by 0.53% and 0.30% in terms of classification accuracy, where the advantage is clearer than the other two cases. It can be obviously found that, with the shrinkage of the training set, the advanced dropout technique maintains the first place on classification accuracy and enlarges the gap with the corresponding second best technique.

In Table 10, the numbers of hidden layers of the FC neural networks are set as 2, 4, and 8. With the increase of the model depth, most of the referred techniques, except for the soft dropouts, performs worse and the maximum drop in classification accuracy is more than 0.5%. Meanwhile, the performance of the base model without any dropout techniques also decreases. However, the advanced dropout technique maintains its classification accuracies of more than 98.8%. In addition, it achieves the best performance among all techniques in each model depth.

In summary, the proposed advanced dropout technique reduces the gap between training and test accuracies (*i.e.*, reducing overfitting) and achieves the highest test accuracies at the same time. Moreover, the advanced dropout technique performs best among the referred methods, when we increase the model depth or decrease the training sample number. It also performs well in the extreme case, for example, $\xi = 15$ or the eight-hidden-layer FC neural network. The experimental results verify the superior capability of overfitting prevention of the advanced dropout technique.

6 Extension of Applications of Advanced Dropout

In this section, we will generalize the advanced dropout as a practical technique in four other application areas, rather than merely a regularization technique for DNN training in image classification. Hence, we apply the advanced dropout technique in uncertainty inference and network pruning in computer vision, text classification, and regression.

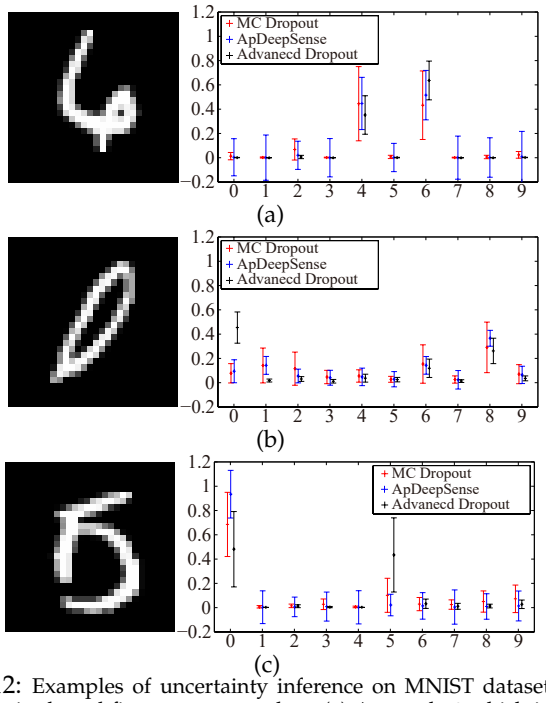


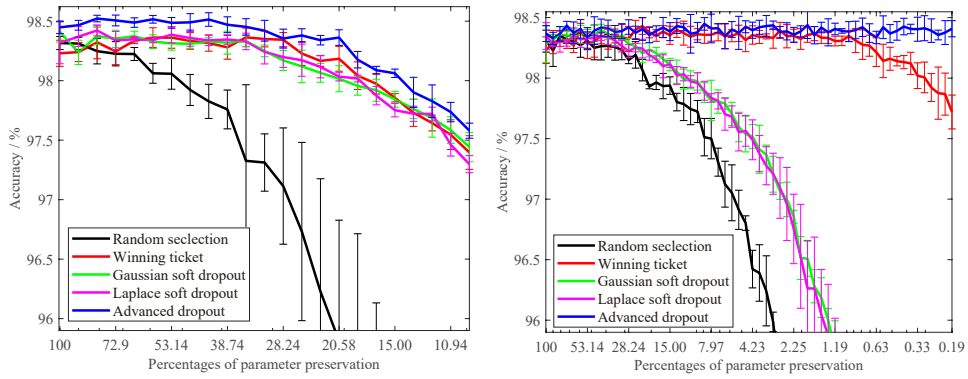
Fig. 12: Examples of uncertainty inference on MNIST dataset. Each column in the subfigures means a class. (a) A sample 6 which is easily confused with 4; (b) a sample 0 which is easily considered as a part of 8; and (c) a sample 5 which is predicted as 0 by all the techniques.

6.1 Uncertainty Inference

Although deep learning has attracted significant attention on research in various fields, such as classification and regression, standard deep learning tools including CNNs and RNNs are not able to capture model uncertainty and confidence through the point estimation in model training by gradient descent-based algorithms [24]. Therefore, we need effective methods, for example, Bayesian learning-based methods, to estimate model uncertainty. In recent years, various techniques have been proposed for this purpose, such as MC dropout [24] and ApDeepSense [54].

In this paper, we propose to implement the advanced dropout technique in uncertainty inference to validate its effectiveness for this propose.

Here, we modify the advanced dropout as a Monte Carlo (MC) dropout technique, perform moment matching in the test process, and estimate the mean $\mathrm{E}[\boldsymbol{y}^*|\boldsymbol{x}^*]$ and the variance $\mathrm{Var}[\boldsymbol{y}^*|\boldsymbol{x}^*]$ of the test output $\boldsymbol{y}^*$ given the corresponding test sample $\boldsymbol{x}^*$ as



(a) Node pruning (b) Parameter pruning

Fig. 13: The performance of the advanced dropout technique for (a) node pruning and (b) parameter pruning compared with the random selection, the winning ticket [55], and Gaussian soft dropout and Laplace soft dropout [15] on the MNIST dataset. Note that the percentages of parameter preservation are displayed on a logarithmic scale.

Algorithm 1 Parameter pruning with the advanced dropout technique

Require: Randomly initialized parameter sets $\boldsymbol{\Theta}_0$ and $\boldsymbol{\Lambda}_0$
Ensure: $\boldsymbol{\Theta}_j$: estimated model parameters in the jth round

1: Initialize a network with parameter sets $\boldsymbol{\Theta}_0$ and $\boldsymbol{\Lambda}_0$.
2: Train the network with the advanced dropout technique and obtain parameter sets $\boldsymbol{\Theta}_j$ and $\boldsymbol{\Lambda}_j$.
3: Prune $q\%$ parameters in $\boldsymbol{\Theta}_j$ which correspond to the $q\%$ largest dropout rates calculated by (27).
4: Reset the remaining parameters in $\boldsymbol{\Theta}_j$ to their values in $\boldsymbol{\Theta}_0$ and parameters in $\boldsymbol{\Lambda}_j$ to their values in $\boldsymbol{\Lambda}_0$.
5: Return to the step 2.

$$\mathrm{E}[\boldsymbol{y}^*|\boldsymbol{x}^*] = \int \boldsymbol{y}^* \int q_{\boldsymbol{\Phi}}(\boldsymbol{W}|\boldsymbol{Z}) p(\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{W}) d\boldsymbol{W} d\boldsymbol{y}^* \approx \frac{1}{T}\sum_{t=1}^{T} \hat{\boldsymbol{y}}^*(\boldsymbol{x}^*, \boldsymbol{W}^t), \tag{28}$$

$$\mathrm{Var}[\boldsymbol{y}^*|\boldsymbol{x}^*] = \int (\boldsymbol{y}^*)^2 \int q_{\boldsymbol{\Phi}}(\boldsymbol{W}|\boldsymbol{Z}) p(\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{W}) d\boldsymbol{W} d\boldsymbol{y}^* - \mathrm{E}^2[\boldsymbol{y}^*|\boldsymbol{x}^*] \approx \frac{1}{T}\sum_{t=1}^{T} \left(\hat{\boldsymbol{y}}^*(\boldsymbol{x}^*, \boldsymbol{W}^t) - \mathrm{E}[\boldsymbol{y}^*|\boldsymbol{x}^*]\right)^2, \tag{29}$$

where T is number of times of the stochastic forward passes through the DNN, and the mean $\mathrm{E}[\boldsymbol{y}^*|\boldsymbol{x}^*]$ and the variance $\mathrm{Var}[\boldsymbol{y}^*|\boldsymbol{x}^*]$ are considered as the expected output and the model uncertainty.

We conducted experiments via a FC neural network with two hidden layers and 800 hidden nodes of each on the MNIST dataset and compared it with two referred methods, *i.e.*, MC dropout [24] and ApDeepSense [54].

The performance of each technique is assessed by area under the receiver operating characteristic curve (AUROC) of two measurements of uncertainty, *i.e.*, max probability $\mathcal{P}$ and entropy $\mathcal{H}$, which are introduced in the works [56], [57]. $\mathcal{P}$ and $\mathcal{H}$ of test output $\boldsymbol{y}^*$ given the corresponding test sample $\boldsymbol{x}^*$ are respectively defined as

$$\mathcal{P} = \max_{c=1,\cdots,C} p(y_c^*|\boldsymbol{x}^*, \boldsymbol{W}), \tag{30}$$

$$\mathcal{H} = -\sum_{c=1}^{C} p(y_c^*|\boldsymbol{x}^*, \boldsymbol{W}) \ln(p(y_c^*|\boldsymbol{x}^*, \boldsymbol{W})), \tag{31}$$

where C is the number of classes. From Table 11, it is observed that the advanced dropout technique surpasses both referred techniques by more than 0.03 and 0.04 on the AUC of max probability and entropy, respectively. Meanwhile, the accuracies of these techniques are shown in Table 11, while the advanced dropout outperforms both referred techniques with a significant improvement around 4%. Furthermore, Figure 11 shows the confusion matrices of these techniques, one subfigure each. The advanced dropout outperforms the other two techniques in eight classes (class 2-9) with the largest increase by about 15% and the smallest one by about 2%, while it keeps the same level as the others in the other two classes.

Figure 12 shows four examples of visualization of the uncertainty inference results predicted by all three techniques. In Figure 12(a), all the three techniques are confused between 4 and 6, even though the ground truth is 6. While the two referred techniques predict the wrong class, the advanced dropout technique makes a correct prediction with the highest mean and the smallest variance. This indicates that it is more confident in the prediction. Similarly in Figure 12(b), the advanced dropout technique predicts the correct class, while the referred techniques are all wrong. Meanwhile, in Figure 12(c), all the three techniques predict to the wrong class 0, while the true label is 5. However, the advanced dropout technique hesitates between the classes 0 and 5 with similar means and larger variances. On the other hand, the referred techniques have their largest prediction values in class 0 with smaller variances than the advanced dropout technique.

In summary, the advanced dropout technique is suitable for model uncertainty inference and performs better than the referred methods. It can effectively infer the uncertainty and improve the classification accuracies simultaneously.

6.2 Network Pruning

In this section, we extend the application of the advanced dropout technique to the field of network pruning, which is a popular and important topic in deep learning. Network pruning strategies can be divided into two categories, *i.e.*, node pruning and parameter pruning. The algorithm for parameter pruning with the advanced dropout technique is summarized in Algorithm 1. The node pruning was conducted in a similar way, which replaces pruning $q\%$ parameters by pruning $q\%$ nodes.

We conducted experiments on the MNIST dataset in an FC neural network with two hidden layers and 800 hidden

TABLE 12: Test accuracies (%) and p-value of Student's t-tests on the Reuters-21578 dataset. Note that the best results are marked in **bold** and the second best results are marked by underline, respectively. The significance level was set as 0.05.

Method	10-1 × 800-10/p-value	10-2 × 800-10/p-value	10-4 × 800-10/p-value
No dropout	$87.68 \pm 0.18/8.12 \times 10^{-7}$	$87.58 \pm 0.24/8.12 \times 10^{-7}$	$87.63 \pm 0.07/8.12 \times 10^{-7}$
Dropout, Bernoulli	$87.75 \pm 0.21/1.45 \times 10^{-6}$	$87.09 \pm 0.72/1.45 \times 10^{-6}$	$87.50 \pm 0.31/1.45 \times 10^{-6}$
Dropout, Gaussian	$87.51 \pm 0.38/6.79 \times 10^{-6}$	$87.45 \pm 0.28/6.79 \times 10^{-6}$	$87.95 \pm 0.20/6.79 \times 10^{-6}$
Dropout, uniform	$88.08 \pm 0.19/4.94 \times 10^{-6}$	$87.83 \pm 0.12/4.94 \times 10^{-6}$	$88.22 \pm 0.30/4.94 \times 10^{-6}$
Concrete dropout	$88.73 \pm 0.18/2.45 \times 10^{-4}$	$\underline{88.82 \pm 0.30}/2.45 \times 10^{-4}$	$88.73 \pm 0.19/2.45 \times 10^{-4}$
Variational dropout	$\underline{88.99 \pm 0.25}/4.32 \times 10^{-3}$	$\underline{88.82 \pm 0.32}/4.32 \times 10^{-3}$	$\underline{88.83 \pm 0.13}/4.32 \times 10^{-3}$
β-dropout	$88.50 \pm 0.25/1.14 \times 10^{-4}$	$88.58 \pm 0.31/1.14 \times 10^{-4}$	$88.67 \pm 0.18/1.14 \times 10^{-4}$
Continuous dropout	$88.11 \pm 0.27/1.96 \times 10^{-5}$	$88.13 \pm 0.18/1.96 \times 10^{-5}$	$88.13 \pm 0.22/1.96 \times 10^{-5}$
Information dropout	$88.77 \pm 0.17/2.63 \times 10^{-4}$	$88.31 \pm 0.28/2.63 \times 10^{-4}$	$88.73 \pm 0.15/2.63 \times 10^{-4}$
Gaussian soft dropout	$88.67 \pm 0.26/4.36 \times 10^{-4}$	$88.63 \pm 0.21/4.36 \times 10^{-4}$	$88.58 \pm 0.21/4.36 \times 10^{-4}$
Laplace soft dropout	$88.69 \pm 0.14/1.05 \times 10^{-4}$	$88.50 \pm 0.21/1.05 \times 10^{-4}$	$88.55 \pm 0.08/1.05 \times 10^{-4}$
Advanced dropout	$\mathbf{89.62 \pm 0.26}$/ N/A	$\mathbf{89.31 \pm 0.11}$/ N/A	$\mathbf{89.35 \pm 0.08}$/ N/A

TABLE 13: Test RMSE and p-value of Student's t-tests on the four UCI datasets. Note that the best results are marked in **bold** and the second best results are marked by underline, respectively. The significance level was set as 0.05.

Dataset Base model Method	Boston Housing 13-2 × 50-1 RMSE/p-value	Concrete Strength 8-2 × 50-1 RMSE/p-value	Wine Quality Red 11-2 × 50-1 RMSE/p-value	Yacht Hydrodynamics 6-2 × 50-1 RMSE/p-value
No dropout	$8.5535 \pm 0.0129/5.67 \times 10^{-5}$	$15.8219 \pm 0.0275/5.92 \times 10^{-4}$	$0.8154 \pm 0.0014/3.51 \times 10^{-3}$	$13.1689 \pm 0.1765/2.05 \times 10^{-4}$
Dropout, Bernoulli	$8.5065 \pm 0.0071/6.88 \times 10^{-5}$	$15.7828 \pm 0.0131/1.69 \times 10^{-4}$	$0.8130 \pm 0.0005/3.86 \times 10^{-3}$	$13.0169 \pm 0.0722/1.11 \times 10^{-5}$
Dropout, Gaussian	$8.5062 \pm 0.0092/2.14 \times 10^{-4}$	$15.7775 \pm 0.0123/1.79 \times 10^{-4}$	$\underline{0.8128 \pm 0.0003}/1.59 \times 10^{-3}$	$12.9740 \pm 0.1621/3.57 \times 10^{-4}$
Dropout, uniform	$8.5446 \pm 0.0143/1.25 \times 10^{-4}$	$15.8134 \pm 0.0244/5.00 \times 10^{-4}$	$0.8135 \pm 0.0004/9.25 \times 10^{-4}$	$13.0386 \pm 0.0791/1.46 \times 10^{-5}$
Concrete dropout	$8.7723 \pm 0.0909/1.14 \times 10^{-3}$	$16.0284 \pm 0.0283/1.14 \times 10^{-5}$	$0.8273 \pm 0.0049/1.69 \times 10^{-3}$	$13.2463 \pm 0.3342/1.74 \times 10^{-3}$
Variational dropout	$8.5078 \pm 0.0090/1.68 \times 10^{-4}$	$15.7920 \pm 0.0162/2.46 \times 10^{-4}$	$0.8133 \pm 0.0004/1.11 \times 10^{-3}$	$13.0626 \pm 0.0592/3.79 \times 10^{-6}$
Gaussian soft dropout	$8.4894 \pm 0.0119/2.93 \times 10^{-3}$	$15.7604 \pm 0.0139/8.43 \times 10^{-4}$	$\underline{0.8128 \pm 0.0003}/2.57 \times 10^{-3}$	$12.3218 \pm 0.0419/3.20 \times 10^{-3}$
Laplace soft dropout	$\underline{8.4728 \pm 0.0074}/7.36 \times 10^{-3}$	$\underline{15.7488 \pm 0.0214}/9.78 \times 10^{-3}$	$0.8133 \pm 0.0005/2.46 \times 10^{-3}$	$\underline{12.2844 \pm 0.0246}/1.88 \times 10^{-3}$
Advanced dropout	$\mathbf{8.4577 \pm 0.0009}$/ N/A	$\mathbf{15.7085 \pm 0.0003}$/ N/A	$\mathbf{0.8119 \pm 0.0001}$/ N/A	$\mathbf{12.2142 \pm 0.0057}$/ N/A

nodes of each. We compared the advanced dropout technique with a random selection method as a baseline, a state-of-the-art technique (winning ticket [55]), and two dropout variants (the Gaussian soft dropout and the Laplace soft dropout [15]). The random selection method stochastically chooses nodes or parameters in each round. We adopted the SGD algorithm for model training and set q as 10.

Figure 13(a) shows the performance of node pruning on the MNIST dataset. The advanced dropout technique achieves higher accuracy than the referred methods on each percentage of node preservation. In addition, Figure 13(b) shows the performance of parameter pruning on the MNIST dataset, which can also demonstrate the better performance of the advanced dropout. Therefore, the advanced dropout technique can be effectively utilized in network pruning and is better than the referred techniques.

6.3 Text Classification

In this section, in addition to the aforementioned computer vision tasks, we further evaluated the performance of the proposed advanced dropout technique in text classification on Reuter-21578 dataset[1]. According to the official data preprocessing procedure, each document in the dataset was expressed by a 10-dimensional vector. The dropout techniques were implemented by using the FC neural networks with one, two, and four hidden layer(s) and 800 hidden nodes each. All the models were trained for 50 epochs using the SGD algorithm with the learning rate as 0.01 and the weight decay as 5×10^{-4}. We evaluated each model for 5 runs with random initializations.

Experimental results and the p-value of the Student's t-tests between the advanced dropout and other referred dropout variants are shown in Table 12. The proposed advanced dropout achieves the best accuracies in the models with different depths. As the p-value are all smaller than the significance level (*i.e.*, $\alpha = 0.05$), the advanced dropout obtains statistically significant improvement.

6.4 Regression

In this section, except for the aforementioned classification tasks, we evaluated the proposed advanced dropout for regression tasks on the UCI datasets [58]. The dropout techniques were implemented by using the FC neural networks with 2 hidden layers and 50 hidden nodes each. All the models were trained for 50 epochs using the SGD algorithm with the learning rate as 0.01 and the weight decay as 5×10^{-4}. We evaluated each model for 5 runs with random initializations. The root mean squared error (RMSE) was chosen as the evaluation metric.

Experimental results and the p-value of the Student's t-tests between the advanced dropout and other referred dropout variants are listed in Table 13. On each dataset, the proposed advanced dropout achieves the smallest RMSEs. As the p-value are all smaller than the significance level (*i.e.*, $\alpha = 0.05$), statistically significant improvement was obtained by the advanced dropout.

7 Conclusions

In this paper, we proposed the advanced dropout technique, a model-free methodology, to improve the ability of overfitting prevention and the classification performance of DNNs. The advanced dropout uses a model-free and easily implemented distribution with a parametric prior to adaptively adjust the dropout rate. Furthermore, the prior parameters are optimized by the SGVB inference to perform an end-to-end training procedure of DNNs. In the experiments, we evaluated the effectiveness of the advanced dropout technique with nine referred techniques in different base models on seven widely used datasets (including five small-scale datasets and two large-scale datasets). The advanced dropout statistically significantly outperformed all the referred techniques. We compared training time and effectiveness ratios between the techniques and found that the advanced dropout achieves highest effectiveness ratios on most datasets. Ablation studies were conducted to analyze the effectiveness of each component. We further compared training time and effectiveness ratios between the techniques and found that the advanced dropout achieves highest effectiveness ratios on most datasets. Next, we conducted a series of analysis of dropout rate characteristics, including convergence of the adaptive dropout rate, the learned distributions of dropout masks, and a comparison with

1. http://kdd.ics.uci.edu/databases/reuters21578/

dropout rate generation without using an explicit distribution. In addition, the ability of overfitting prevention of the advanced dropout was evaluated and confirmed. Finally, we extended the application of the advanced dropout to uncertainty inference, network pruning, text classification, and regression, and we found that the advanced dropout is superior to the corresponding referred methods.

ACKNOWLEDGMENTS

This work was supported in part by the National Key R&D Program of China under Grant 2019YFF0303300 and under Subject II No. 2019YFF0303302, in part by National Natural Science Foundation of China (NSFC) No. 61922015, 61773071, U19B2036, in part by Beijing Natural Science Foundation Project No. Z200002, in part by the Beijing Academy of Artificial Intelligence (BAAI) under Grant BAAI2020ZJ0204, and in part by the Beijing Nova Programme Interdisciplinary Cooperation Project under Grant Z191100001119140.

REFERENCES

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[3] G. Huang, Z. Liu, L. V. Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Computer Vision and Pattern Recognition*, 2017, pp. 2261–2269.

[4] X. Li, L. Yu, D. Chang, Z. Ma, and J. Cao, "Dual cross-entropy loss for small-sample fine-grained vehicle classification," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4204–4212, 2019.

[5] X. Li, D. Chang, T. Tian, and J. Cao, "Large-margin regularized softmax cross-entropy loss," *IEEE Access*, vol. 7, pp. 19572–19578, 2019.

[6] P. Xu, Q. Yin, Y. Qi, Y.-Z. Song, Z. Ma, L. Wang, and J. Guo, "Instance-level coupled subspace learning for fine-grained sketch-based image retrieval," in *European Conference on Computer Vision Workshops*. Springer, 2016, pp. 19–34.

[7] P. Xu, Y. Huang, T. Yuan, K. Pang, Y. Z. Song, T. Xiang, T. M. Hospedales, Z. Ma, and J. Guo, "SketchMate: Deep hashing for million-scale human sketch retrieval," in *Computer Vision and Pattern Recognition*, 2018, pp. 8090–8098.

[8] P. Xu, Q. Yin, Y. Huang, Y. Z. Song, Z. Ma, L. Wang, T. Xiang, W. B. Kleijn, and J. Guo, "Cross-modal subspace learning for fine-grained sketch-based image retrieval," *Neurocomputing*, vol. 278, pp. 75–86, 2018.

[9] Z. Ma, Y. Ding, S. Wen, J. Xie, Y. Jin, Z. Si, and H. Wang, "Shoeprint image retrieval with multi-part weighted CNN," *IEEE ACCESS*, vol. 7, pp. 59728 – 59736, 2019.

[10] Z. Xu, S. Huang, Y. Zhang, and D. Tao, "Webly-supervised fine-grained visual categorization via deep domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1100–1113, 2018.

[11] E. Adeli, K. Thung, L. An, G. Wu, F. Shi, T. Wang, and D. Shen, "Semi-supervised discriminative classification robust to sample-outliers and feature-noises," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 515–522, 2019.

[12] K. Ma, Z. Duanmu, Z. Wang, Q. Wu, W. Liu, H. Yong, H. Li, and L. Zhang, "Group maximum differentiation competition: Model comparison with few samples," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[13] F. Zhu, Z. Ma, X. Li, G. Chen, J.-T. Chien, J.-H. Xue, and J. Guo, "Image-text dual neural network with decision strategy for small-sample image classification," *Neurocomputing*, vol. 328, pp. 182–188, 2019.

[14] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv*, 2012.

[15] J. Xie, Z. Ma, G. Zhang, J.-H. Xue, Z.-H. Tan, and J. Guo, "Soft dropout and its variational Bayes approximation," in *IEEE International Workshop on Machine Learning for Signal Processing*, 2019.

[16] A. Achille and S. Soatto, "Information dropout: Learning optimal representations through noisy computation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2897–2905, 2018.

[17] A. Labach and H. Salehinejad, "Survey of dropout methods for deep neural networks," *arXiv*, pp. 1–11, 2019.

[18] X. Shen, X. Tian, T. Liu, F. Xu, and D. Tao, "Continuous dropout," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 9, pp. 3926–3937, 2018.

[19] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," in *Neural Information Processing Systems*, 2015, vol. 28, pp. 2575–2583.

[20] L. Wan, M. D. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus, "Regularization of neural networks using DropConnect," in *International Conference on Machine Learning*, 2013, pp. 1058–1066.

[21] S. Wang, T. Zhou, and J. A. Bilmes, "Jumpout: Improved dropout for deep neural networks with ReLUs," in *International Conference on Machine Learning*, 2019, pp. 6668–6676.

[22] J. Ba and B. J. Frey, "Adaptive dropout for training deep neural networks," in *Neural Information Processing Systems*, 2013, pp. 3084–3092.

[23] S. Maeda, "A bayesian encourages dropout," in *International Conference on Learning Representation*, 2015.

[24] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: representing model uncertainty in deep learning," in *International Conference on Machine Learning*, 2016, pp. 1050–1059.

[25] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Neural Information Processing Systems*, 2016, pp. 1027–1035.

[26] S. H. Khan, M. Hayat, and F. Porikli, "Regularization of deep neural networks with spectral dropout," *Neural Networks*, vol. 110, pp. 82–90, 2019.

[27] B. Ko, H. Kim, K. Oh, and H. Choi, "Controlled dropout: A different approach to using dropout on deep neural network," in *International Conference on Big Data and Smart Computing*, 2017, pp. 358–362.

[28] Y. Li and Y. Gal, "Dropout inference in bayesian neural networks with alpha-divergences," in *International Conference on Machine Learning*, 2017, pp. 2052–2061.

[29] H. Salehinejad and S. Valaee, "Ising-dropout: A regularization method for training and compression of deep neural networks," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3602–3606.

[30] J. Chen, Z. Wu, J. Zhang, and F. Li, "Mutual information-based dropout: Learning deep relevant feature representation architectures," *Neurocomputing*, vol. 361, pp. 173–184, 2019.

[31] S. I. Wang and C. D. Manning, "Fast dropout training," in *International Conference on Machine Learning*, 2013, pp. 118–126.

[32] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[33] Y. Gal, J. Hron, and A. Kendall, "Concrete dropout," in *Neural Information Processing Systems*, 2017, pp. 3581–3590.

[34] L. Liu, Y. Luo, X. Shen, M. Sun, and B. Li, "β-dropout: A unified dropout," *IEEE Access*, vol. 7, pp. 36140–36153, 2019.

[35] S. R. Bulò, L. Porzi, and P. Kontschieder, "Dropout distillation," in *International Conference on Machine Learning*, 2016.

[36] X. Ma, Y. Gao, Z. Hu, Y. Yu, Y. Deng, and E. Hovy, "Dropout with expectation-linear regularization," in *International Conference on Learning Representations*, 2017.

[37] H. Gao, J. Pei, and H. Huang, "Demystifying dropout," in *International Conference on Machine Learning*, 2019.

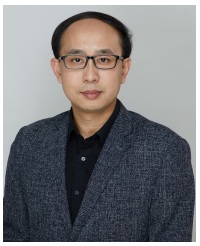
[38] Z. Li, B. Gong, and T. Yang, "Improved dropout for shallow and deep learning," in *Advances in Neural Information Processing Systems*, 2016.

[39] H. Wang, W. Yang, Z. Zhao, T. Luo, J. Wang, and Y. Tang, "Rademacher dropout: An adaptive dropout for deep neural network via optimizing generalization gap," *Neurocomputing*, vol. 357, pp. 177–187, 2019.

[40] Y. Liu, W. Dong, L. Zhang, D. Gong, and Q. Shi, "Variational Bayesian dropout with a hierarchical prior," in *Computer Vision and Pattern Recognition*, 2019, pp. 7124–7133.

[41] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *International Conference on Learning Representations*, 2014.
[42] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer Science+Business Media LLC., 2006.
[43] Y. LeCun and C. Cortes, "The MNIST database of handwritten digits," http://yann.lecun.com/exdb/mnist/, 1998.
[44] A. Krizhevsky, "Learning multiple layers of features from tiny images," techreport, CIFAR, 2009.
[45] O. Vinyals, C. Blundell, T. P. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," *arXiv*, 2016.
[46] G. Griffin, A. Holub, and P. Perona, "Caltech256 image dataset," http://www.vision.caltech.edu/Image_Datasets/Caltech256/, 2006.
[47] A. V. Den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," *arXiv*, 2016.
[48] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Feifei, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
[49] S. Li, S. Y. Siu, T. Fang, and L. Quan, "Efficient multi-view surface refinement with adaptive resolution control," in *European Conference on Computer Vision*, 2016, pp. 349–364.
[50] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv*, 2017.
[51] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *British Machine Vision Conference*, 2016.
[52] Z. Ma and A. Leijon, "Bayesian estimation of Beta mixture models with variational inference," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2160–2173, 2011.
[53] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," in *International Conference on Learning Representations*, 2017.
[54] S. Yao, Y. Zhao, H. Shao, C. Zhang, A. Zhang, D. Liu, S. Liu, L. Su, and T. Abdelzaher, "ApDeepSense: Deep learning uncertainty estimation without the pain for IoT applications," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, 2018, pp. 334–343.
[55] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," in *International Conference on Learning Representations*, 2019.
[56] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *International Conference on Learning Representations*, 2017.
[57] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," in *Neural Information Processing Systems*, 2018, pp. 7047–7058.
[58] D. Dua and C. Graff, "UCI machine learning repository," http://archive.ics.uci.edu/ml, 2017.

Jiyang Xie received his B.E. degree in information engineering from Beijing University of Posts and Telecommunications (BUPT), China, in 2017, where he is currently pursuing the Ph.D. degree. His research interests include pattern recognition and machine learning fundamentals with a focus on applications in image processing, data mining, and deep learning.



Zhanyu Ma is currently a Professor at Beijing University of Posts and Telecommunications, Beijing, China, since 2019. He received the Ph.D. degree in electrical engineering from the KTH-Royal Institute of Technology, Sweden, in 2011. From 2012 to 2013, he was a Postdoctoral Research Fellow with the School of Electrical Engineering, KTH-Royal Institute of Technology. He has been an Associate Professor with the Beijing University of Posts and Telecommunications, Beijing, China, from 2014 to 2019. His research interests include pattern recognition and machine learning fundamentals with a focus on applications in computer vision, multimedia signal processing, data mining. He is a Senior Member of IEEE.

Jianjun Lei (M'11-SM'17) received the Ph.D. degree in signal and information processing from Beijing University of Posts and Telecommunications, Beijing, China, in 2007. He was a visiting researcher at the Department of Electrical Engineering, University of Washington, Seattle, WA, from August 2012 to August 2013. He is currently a Professor at Tianjin University, Tianjin, China. His research interests include 3D video processing, virtual reality, and artificial intelligence.

Guoqiang Zhang received the B. Eng. from University of Science and Technology of China (USTC) in 2003, M.Phil. degree from University of Hong Kong in 2006, and Ph.D. degree from Royal Institute of Technology in 2010. From the spring of 2011, he worked as a Postdoctoral Researcher at Delft University of Technology. From the spring of 2015, he worked as a senior researcher at Ericsson AB, Sweden. Since 2017, he has been a senior lecturer in the School of Electrical and Data Engineering, University of Technology Sydney, Australia. He is an IEEE member. His current research interests include distributed optimization, large scale optimization, deep learning, and multimedia signal processing.

Jing-Hao Xue received the Dr. Eng. degree in signal and information processing from Tsinghua University in 1998 and the Ph.D. degree in statistics from the University of Glasgow in 2008. He is an associate professor in the Department of Statistical Science, University College London. His research interests include statistical machine learning, high-dimensional data analysis, pattern recognition and image analysis.

Zheng-Hua Tan (M'00–SM'06) received the B.Sc. and M.Sc. degrees in electrical engineering from Hunan University, Changsha, China, in 1990 and 1996, respectively, and the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 1999.

He is a Professor and a Co-Head of the Centre for Acoustic Signal Processing Research at Aalborg University, Aalborg, Denmark. He was a Visiting Scientist at the Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, USA, an Associate Professor at Shanghai Jiao Tong University, and a postdoctoral fellow at KAIST, Daejeon, Korea. His research interests include machine learning, deep learning, pattern recognition, speech and speaker recognition, noise-robust speech processing, multimodal signal processing, and social robotics. He has authored/co-authored over 200 publications in refereed journals and conference proceedings. He is the Chair of the IEEE Signal Processing Society Machine Learning for Signal Processing Technical Committee (MLSP TC). He is an Associate Editor for IEEE/ACM Transactions on Audio, Speech and Language Processing. He was the General Chair for IEEE MLSP 2018 and a TPC co-chair for IEEE SLT 2016.

Jun Guo received the B.E. and M.E. degrees from the Beijing University of Posts and Telecommunications (BUPT), China, in 1982 and 1985, respectively, and the Ph.D. degree from the Tohuku Gakuin University, Japan, in 1993. He is currently a professor and a vice president with BUPT. He has authored over 200 papers in journals and conferences, including Science, Nature Scientific Reports, the IEEE Transactions on PAMI, Pattern Recognition, AAAI, CVPR, ICCV, and SIGIR. His research interests include pattern recognition theory and application, information retrieval, content-based information security, and bioinformatics.