Distributionally Robust Ensemble of Lottery Tickets Towards Calibrated Sparse Network Training

Anonymous Author(s)
Affiliation
Address
email

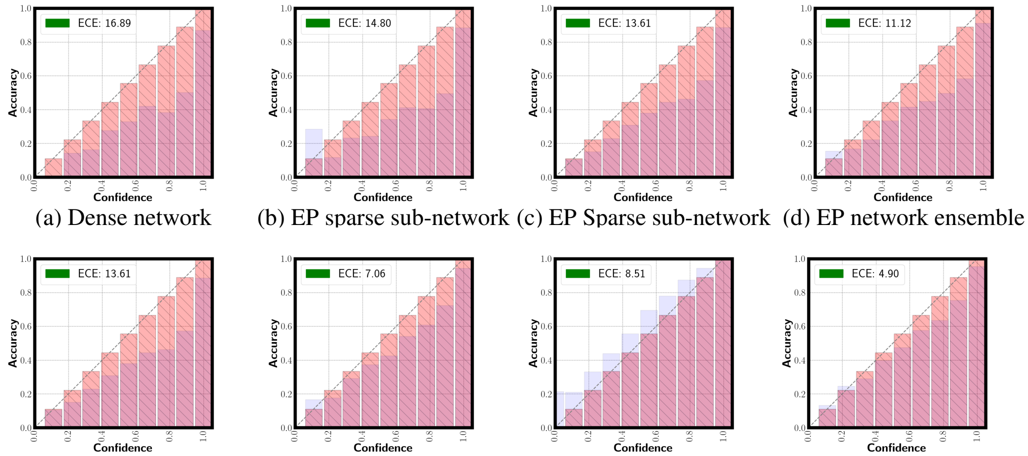
Abstract

The recently developed sparse network training methods, such as Lottery Ticket Hypothesis (LTH) and its variants, have shown impressive learning capacity by finding sparse sub-networks from a dense one. While these methods could largely sparsify deep networks, they generally focus more on realizing comparable accuracy to dense counterparts yet neglect network calibration. However, how to achieve calibrated network predictions lies at the core of improving model reliability, especially when it comes to addressing the overconfident issue and out-of-distribution cases. In this study, we propose a novel Distributionally Robust Optimization (DRO) framework to achieve an ensemble of lottery tickets towards calibrated network sparsification. Specifically, the proposed DRO ensemble aims to learn multiple diverse and complementary sparse sub-networks (tickets) with the guidance of uncertainty sets, which encourage tickets to gradually capture different data distributions from easy to hard and naturally complement each other. We theoretically justify the strong calibration performance by showing how the proposed robust training process guarantees to lower the confidence of incorrect predictions. Extensive experimental results on several benchmarks show that our proposed lottery ticket ensemble leads to a clear calibration improvement without sacrificing accuracy and burdening inference costs. Furthermore, experiments on OOD datasets demonstrate the robustness of our approach in the open-set environment.

1 Introduction

While there is remarkable progress in developing deep neural networks with densely connected layers, most of these dense networks have poor calibration performance [9], limiting their applicability in safety-critical domains like self-driving cars [3] and medical diagnosis [11]. The poor calibration is mainly due to the fact that there exists a good number of wrongly classified data samples (*i.e.,* low accuracy) with high confidence resulting from the memorization effect introduced by an over-parameterized architecture [24]. Recent sparse network training methods, such as Lottery Ticket Hypothesis (LTH) [6] and its variants [2, 32, 17, 15, 30] generally assume that there exists a sparse sub-network (*i.e.*, lottery ticket) in a randomly initialized dense network, which could be trained in isolation and also match the performance of its dense counterpart network in terms of accuracy. While these methods may, to some extent, alleviate the overconfident issue, two key challenges remain to be addressed: (i) most of sparse network training methods require pre-training of a dense network followed by multi-step iterative pruning, making the overall training process highly costly, especially for large dense networks; (ii) even for techniques that do not rely on pre-training and iterative pruning (*e.g.,* Edge Popup or EP [23]), their learning goal focuses on pushing the accuracy up to the original dense networks and hence may still exhibit a severely over-fitting behavior, leading to a poor calibration performance as demonstrated in Figure 1 (b).

Inspired by the recent success of using ensembles to estimate uncertainties [13, 29], a potential solution to realize well-calibrated predictions would be training multiple sparse sub-networks and



(a) Dense network (b) EP sparse sub-network (c) EP Sparse sub-network (d) EP network ensemble

(e) Sparse sub-network 1 (f) Sparse sub-network 2 (g) Sparse sub-network 3 (h) DRO ensemble of (e-g)

Figure 1: Calibration performance by expected calibration error (ECE) on Cifar100 dataset with ResNet101 architecture with density $\mathcal{K}=15\%$. EP refers to the Edge Popup algorithm [23].

building an ensemble from them. As such, by leveraging accurate uncertainty quantification, the ensemble is expected to achieve better calibration. However, existing ensemble models of sparse networks rely on pre-training and iterative fine-tuning for learning each sub-network [17, 30], leading to a significant overhead for building the entire ensemble. Furthermore, an ensemble of independently trained sparse sub-networks does not necessarily improve the calibration performance. Since these networks are trained in a similar fashion from the same training data distribution, they could be strongly correlated such that the ensemble model will potentially inherit the overfitting behavior of each sub-network as shown in Figure 1(c). Therefore, the calibration capacity of sparse sub-network ensemble can be compromised as shown empirically in Figure 1 (d).

To further enhance the calibration of the ensemble, it is critical to ensure sufficient diversity among sparse sub-networks so that they are able to complement each other. One natural way to achieve diversity is to allow each sparse sub-network (ticket) to primarily focus on a specific part of training data distribution. This inspires us to leverage the AdaBoost [25] framework that sequentially finds tickets by manipulating training data distribution based on errors. By this means, the AdaBoost facilitates the training for a sequence of complementary sparse sub-networks. However, the empirical analysis (see Table 1) reveals that in the AdaBoost ensemble, most sub-networks (except for the first one) severely under-fit data leading to poor generalization ability. This is mainly because of the overfitting behavior of the first sub-network, which assigns very low training losses to the majority of data samples, making the subsequent sub-networks concentrate on very rare difficult samples that are likely to be outliers or noises. Hence, directly learning from these difficult samples without having global knowledge of the entire training distribution will result in the failure of subsequent training tickets and also hurt the overall calibration.

To this end, we need a more robust learning process for proper training of complementary sparse sub-networks, each of which can be learned in an efficient way to ensure the cost-effective construction of the entire ensemble. We propose a Distributionally Robust Optimization (DRO) framework to schedule learning an ensemble of lottery tickets (sparse sub-networks) with complimentary calibration behaviors that contribute to an overall well-calibrated ensemble as shown in Figure 1 (e-h). Our technique directly searches sparse sub-networks in a randomly initialized dense network without pre-training or iterative pruning. Unlike the AdaBoost ensemble, the proposed ensemble ticket method starts from the original training distribution and eventually allows learning each sub-network from different parts of the training distribution to enrich diversity. This is also fundamentally different from existing sparse ensemble models [17, 30], which attempt to obtain diverse sub-networks in a heuristic way by relying on different learning rates. As a result, these models offer no guaranteed complementary behavior among sparse sub-networks to cover a different part of training data, which is essential to alleviate the overfitting behavior of the learned sparse sub-networks. In contrast, we realize a principled scheduling process by changing the uncertainty set of DRO, where a small set pushes sub-networks learning with easy data samples and a large set focuses on the difficult ones (see Figure 2). By this means, the ticket ensemble governed by our DRO framework could work complementary and lead to much better calibration ability as demonstrated in Figure 1(h). On the one hand, we hypothesize that the ticket found with easy data samples will tend to be learned and overfitted easily, resulting in overconfident predictions (Figure 1(e)). On the other hand, the ticket focused on more difficult data samples will be less likely to overfit and may become conservative and give under-confident predictions. Thus, it is natural to form an ensemble of such lottery tickets to complement each other in making calibrated predictions. As demonstrated in Figure 1 (h), owing to the diversity in the sparse sub-networks (e-g), the DRO ensemble exhibits better calibration ability. It is also worth noting that under the DRO framework, our sparse sub-networks already improve the calibration ability as shown in Figure 1 (f-g), which is further confirmed by our theoretical results.

Experiments conducted on three benchmark datasets demonstrate the effectiveness of our proposed technique compared to sparse counterparts and dense networks. Furthermore, we show through the experimentation that because of the better calibration, our model is being able to perform well on the distributionally shifted datasets [6] (CIFAR10-C and CIFAR100-C). The experiments also demonstrate that our proposed DRO ensemble framework can better detect open-set samples on varying confidence thresholds. The contribution of this work can be summarized as follows:

- a new sparse ensemble framework that combines multiple sparse sub-networks to achieve better calibration performance without dense network training and iterative pruning.
- a distributionally robust optimization framework that schedules the learning of an ensemble complementary sub-networks (tickets),
- theoretical justification of the strong calibration performance by showing how the proposed robust training process guarantees to lower the confidence of incorrect predictions in Theorem 2,
- extensive empirical evidence on the effectiveness of the proposed lottery ticket ensemble in terms of competitive classification accuracy and improved open-set detection performance.

2 Related Work

Sparse networks training. Sparse network training has received increasing attention in recent years. Representative techniques include lottery ticket hypothesis (LTH) [6] and its variants [4, 28]. To avoid training a dense network, supermasks have been used to find the winning ticket in the dense network without training network weights [32]. Edge-Popup (EP) extends this idea by leveraging training scores associated with the neural network weights and only weights with top scores are used for predictions. There are two key limitations to most existing LTH techniques. First, most of them require pre-training of a dense network followed by multi-step iterative pruning making the overall training process expensive. Second, their learning objective remains as improving the accuracy up to the original dense networks and may still suffer from over-fitting (as shown in Figure 1).

Sparse network ensemble. There are recent advancements in building ensembles from sparse networks. A pruning and regrowing strategy has been developed in a model, called CigL [15], where dropout serves as an implicit ensemble to improve the calibration performance. CigL requires weight updates and performs pruning and growing for multiple rounds, leading to a high training cost. Additionally, dropping many weights may lead to a performance decrease, which prevents building highly sparse networks. This idea has been further extended by using different learning rates to generate different typologies of the network structure for each sparse network [17, 30]. While diversity among sparse networks can be achieved, there is no guarantee that this can improve the calibration performance of the final ensemble. In fact, different networks may still learn from the training data in a similar way. Hence, the learned networks may exhibit similar overfitting behavior with a high correlation, making it difficult to generate a well-calibrated ensemble. In contrast, the proposed DRO ensemble schedules different sparse networks to learn from complementary parts of the training distribution, leading to improved calibration with theoretical guarantees.

Model calibration. Various attempts have been proposed to make the deep models more reliable either through calibration [9, 22, 28] or uncertainty quantification [7, 26]. Post-calibration techniques have been commonly used, including temperature scaling [22, 9], using regularization to penalize overconfident predictions [21]. Recent studies show that post-hoc calibration falls short of providing reliable predictions [20]. Most existing techniques require additional post-processing steps and an additional validation dataset. In our setting, we aim to improve the calibration ability of sparse networks without introducing additional post-calibration steps or validation dataset.

3 Methodology

Let $\mathcal{D}_{\mathcal{N}} = \{\mathbf{X}, \mathbf{Y}\} = \{(\mathbf{x}_1, y_1), .., (\mathbf{x}_N, y_N)\}$ be a set of training samples where each $\mathbf{x}_n \in \mathbb{R}^D$ is a D-dimensional feature vector and $y_n \in [1, C]$ be associated label with C total classes. Let M be

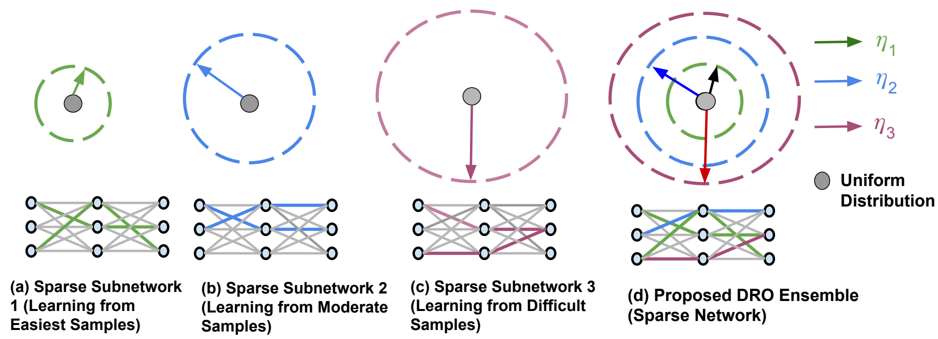


Figure 2: Robust ensemble where η defines the size of an uncertainty set with $\eta_1 \leq \eta_2 \leq \eta_3$.

the total number of base learners used in the given ensemble technique. Further, consider $\mathcal{K}$ to be the density ratio in the given network, which denotes the percentage of weights we keep during the training process. The major notations are summarized in the Appendix.

3.1 Preliminaries

Edge-Popup (EP) [23]. EP finds a lottery ticket (sparse sub-network) from a randomly initialized dense network based on the score values learned from training data. Specifically, to find the sub-network with density $\mathcal{K}$, the algorithm optimizes the scores associated with each weight in the dense network. During the forward pass, the top-$\mathcal{K}$ weights in each layer are selected based on their scores. During the backward pass, scores associated with all weights are updated, which allows potentially useful weights that are ignored in previous forward passes to be re-considered.

Expected calibration error. Expected Calibration Error (ECE) measures the correspondence between predicted probability and empirical accuracy [18]. Specifically, mis-calibration is computed based on the difference in expectation between confidence and accuracy: $\mathbb{E}_{\hat{p}}\left[|\mathbb{P}(\hat{y}=y|\hat{p}=p)-p|\right]$. In practice, we approximate the expectation by partitioning confidences into T bins (equally spaced) and take the weighted average on the absolute difference between each bins' accuracy and confidence. Let B_t denote the t-th beam and we have ECE $=\sum_{t=1}^{T}\frac{|B_t|}{N}|acc(B_t)-conf(B_t)|$.

3.2 Distributionally Robust Ensemble (DRE)

As motivated in the introduction, to further enhance the calibration of a deep ensemble, it is instrumental to introduce sufficient diversity among the component sparse sub-networks so that they can complement each other when forming the ensemble. One way to achieve diversity is to allow each sparse sub-network to primarily focus on a specific part of the training data distribution. Figure 2 provides an illustration of this idea, where the training data can be imagined to follow a multivariate Gaussian distribution with the red dot representing its mean. In this case, the first sub-network will learn the most common patterns by focusing on the training data close to the mean. The subsequent sub-networks will then learn relatively rare patterns by focusing on other parts of the training data (*e.g.,* two or three standard deviations from the mean).

AdaBoost ensemble. The above idea inspires us to leverage the AdaBoost framework [25] to manipulate the training distribution that allows us to train a sequence of complementary sparse sub-networks. In particular, we train the first sparse sub-network from the original training distribution, where each data sample has an equal probability to be sampled. In this way, the first sparse sub-network can learn the common patterns from the most representative training samples. Starting from the second sub-network, the training distribution is changed according to the losses suffered from the previous sub-network during the last round of training. This allows the later sub-networks to focus on the difficult data samples by following the spirit of AdaBoost.

However, our empirical results reveal that in the AdaBoost ensemble, most sub-networks (except for the first one) severely underfit the training data, leading to a rather poor generalization capability. This is caused by the overfitting behavior of the first sparse sub-network, which assigns very small training losses to a majority of data samples. As a result, the subsequent sub-networks can only focus on a limited number of training samples that correspond to relatively rare patterns (or even outliers and noises) in the training data. Directly learning from these difficult data samples without a general knowledge of the entire training distribution will result in the failure of training the sub-networks.

Distributionally robust ensemble (DRE). To tackle the challenge as outlined above, we need a more robust learning process to ensure proper training of complementary sparse sub-networks. Different from the AdaBoost ensemble, the training of all sub-networks starts from the original training distribution in the DRO framework. Meanwhile, it also allows each sub-network to eventually focus on learning from different parts of the training distribution to ensure the desired diverse and complementary behavior. Let $l(\mathbf{x}_n, \Theta)$ denote the loss associated with the n^{th} data sample with Θ being the parameters in the sparse sub-network. Then, the total loss is given by

$$\mathcal{L}^{\text{Robust}}(\Theta) = \max_{\mathbf{z} \in \mathcal{U}^{\text{Robust}}} \sum_{n=1}^{N} z_n l(\mathbf{x}_n, \Theta) \tag{1}$$

The uncertainty set defined to assign weights $\mathbf{z}$ is given as

$$\mathcal{U}^{\text{Robust}} := \left\{ \mathbf{z} \in \mathbb{R}^N : \mathbf{z}^\top \mathbf{1} = 1, \mathbf{z} \geq 0, D_f(\mathbf{z} \| \frac{\mathbf{1}}{N}) \leq \eta \right\} \tag{2}$$

where $D_f(\mathbf{z}\|\mathbf{q})$ is f-divergence beCitween two distributions $\mathbf{z}$ and $\mathbf{q}$ and η controls the size of the uncertainty set and $\mathbf{1} \in 1^N$ is N-dimensional unit vector. Depending on the η value, the above robust framework instantiates different sub-networks. For example, by making $\eta \to \infty$, we have $\mathcal{U}^{\text{Robust}} = \left\{ \mathbf{z} \in \mathbb{R}^N : \mathbf{z}^\top \mathbf{1} = 1, \mathbf{z} \geq 0, D_f(\mathbf{z} \| \frac{\mathbf{1}}{N}) \leq \infty \right\}$. In this case, we train a sub-network by only using the most difficult sample in the training set. On the other extreme with $\eta \to 0$, we have $\mathcal{U}^{\text{Robust}} = \left\{ \mathbf{z} \in \mathbb{R}^N : \mathbf{z}^\top \mathbf{1} = 1, \mathbf{z} \geq 0, D_f(\mathbf{z} \| \frac{\mathbf{1}}{N}) \leq 0 \right\}$, which assigns equal weights to all data samples. So, the sub-network learns from the original training distribution.

To fully leverage the key properties of the robust loss function as described above, we propose to perform distributionally robust ensembling learning to generate a diverse set of sparse sub-networks with well-controlled overfitting behavior that can collectively achieve superior calibration performance. The training process starts with a relatively small η value to ensure that the initially generated sub-networks can adequately capture the general patterns from the most representative data samples in the original training distribution. The training proceeds by gradually increasing the η value, which allows the subsequent sub-networks to focus on relatively rare and more difficult data samples. As a result, the later generated sub-networks tend to produce less confident predictions that complement the sub-networks generated in the earlier phase of the training process. This diverse and complementary behavior among different sparse sub-networks is clearly illustrated in Figure 1 (e)-(g). During the ensemble phase, we combine the predictions of different sub-networks in the logit space by taking the mean and then performing the softmax. In this way, the sparse sub-networks with high η values help to lower the overall confidence score, especially those wrongly predicted data samples. Furthermore, the sub-networks with lower η values help to bring up the confidence score of correctly predicted data samples. Thus, the overall confidence score will be well compensated, resulting in a better calibrated ensemble.

3.3 Theoretical Analysis

In this section, we theoretically justify why the proposed DRE framework improves the calibration performance by extending the recently developed theoretical framework on multi-view learning [1]. In particular, we will show how it can effectively lower the model's false confidence on its wrong predictions resulting from spurious correlations. For this, we first define the problem setup that includes some key concepts used in our theoretical analysis. We then formally show that DRO helps to decorrelate the spurious correlation by learning from less frequent features that characterize difficult data samples in a training dataset. This important property further guarantees better calibration performance of DRO as we show in the main theorem.

Problem setup. Assume that each data sample $\mathbf{x}_n \in \mathbb{R}^D$ is divided into P total patches, where each patch is a d-dimensional vector. For the sake of simplicity, let us assume each class $c \in [1, C]$ has two characterizing (major) features $\mathbf{v}_c = \{\mathbf{v}_{c,l}\}_{l=1}^{L}$ with $L = 2$. For example, the features for `Cars` could be `Headlights` and `Tires`. Let $\mathcal{D}_N^S$ and $\mathcal{D}_N^M$ denote the set of *single-view* and *multi-view* data samples, respectively, which are formally defined as

$$\begin{cases} \{\mathbf{x}_n, y_n\} \in \mathcal{D}_N^S \text{ if one of } \mathbf{v}_{c,1} \text{ or } \mathbf{v}_{c,2} \text{ appears along with some noise features} \\ \{\mathbf{x}_n, y_n\} \in \mathcal{D}_N^M \text{ if both } \mathbf{v}_{c,1} \text{ and } \mathbf{v}_{c,2} \text{ appears along with some noise features} \end{cases} \tag{3}$$

The noise features (also called minor features) refer to those that do not characterize (or differentiate) a given class c (*e.g.,* being part of the background). In important applications like computer vision, images supporting such a "multi-view" structure is very common [1]. For example, for most car images, we can observe all main features, such as `Wheels`, `Tires`, and `Headlights` so they belong to $\mathcal{D}_N^M$. Meanwhile, there may also be images, where multiple features are missing. For example, if the car image is taken from the front, the tire and wheel features may not be captured. In most real-world datasets, such single-view data samples are usually much limited as compared to their multi-view counterparts. The Appendix provides concrete examples of both single and multi-view images. Let us consider $(\mathbf{x}, y) \in \mathcal{D}_N^S$ with the major feature $\mathbf{v}_{c,l}$ where $y = c$. Then each patch $\mathbf{x}^p \in \mathbb{R}^d$ can be expressed as

$$\mathbf{x}^p = a^p \mathbf{v}_{c,l} + \sum_{\mathbf{v}' \in \cup \setminus \mathbf{v}_c} \alpha^{p,\mathbf{v}'} \mathbf{v}' + \epsilon^p \tag{4}$$

where $\cup = \{\mathbf{v}_{c,1}, \mathbf{v}_{c,2}\}_{c=1}^C$ is collection of all features, $a^p > 0$ is the weight allocated to feature $\mathbf{v}_{c,l}$, $\alpha^{p,\mathbf{v}'} \in [0, \gamma]$ is the weight allocated to the noisy feature $\mathbf{v}'$ that is not present in feature set $\mathbf{v}_c$ *i.e.,* $\mathbf{v}' \in \cup \setminus \mathbf{v}_c$, and $\boldsymbol{\epsilon}^p \sim \mathcal{N}(0, (\sigma^p)^2 \mathbb{1})$ is a random Gaussian noise. In (4), a patch $\mathbf{x}^p$ in a single-view sample $\mathbf{x}$ also contains set of minor (noise) features presented from other classes *i.e.,* $\mathbf{v}' \in \cup \setminus \mathbf{v}_c$ in addition to the main feature $\mathbf{v}_{c,l}$. Since $\mathbf{v}_{c,l}$ characterizes class c, we have $a^p > \alpha^{p,\mathbf{v}'}; \forall \mathbf{v}' \in \cup \setminus \mathbf{v}_c$. However, since the single-view data samples are usually sparse in the training data, it may prevent the model from accumulating a large a^p for $\mathbf{v}_{c,l}$ as shown Lemma 1 below. In contrast, some noise $\mathbf{v}'$ may be selected as the dominant feature (due to spurious correlations) to minimize the errors of specific training samples, leading to potential overfitting of the model.

We further assume that the network contains H convolutional layers, which outputs $F(\mathbf{x}; \Theta) = (F_1(\mathbf{x}), ...F_C(\mathbf{x})) \in \mathbb{R}^C$. The logistic output for the c^{th} class can be represented as

$$F_c(\mathbf{x}) = \sum_{h \in [H]} \sum_{p \in [P]} \texttt{ReLU}[\langle \Theta_{c,h}, \mathbf{x}^p \rangle] \tag{5}$$

where $\Theta_{c,h}$ denote the h^{th} convolution layer (feature map) associated with class c. Under the above data and network setting, we propose the following lemma.

Lemma 1. *Let $\mathbf{v}_{c,l}$ be the main feature vector present in the single-view data $\mathcal{D}_N^S$. Assume that number of single-view data samples containing feature $\mathbf{v}_{c,l}$ is limited as compared with the rest,* i.e., *$N_{\mathbf{v}_{c,l}} \ll N_{\cup \setminus \mathbf{v}_{c,l}}$. Then, at any iteration $t > 0$, we have*

$$\langle \Theta_{c,h}^{t+1}, \mathbf{v}_{c,l} \rangle = \langle \Theta_{c,h}^t, \mathbf{v}_{c,l} \rangle + \beta \max_{\mathbf{z} \in \mathcal{U}} \sum_{n=1}^N z_n \left[\mathbb{1}_{y_j = c} (V_{c,h,l}(\mathbf{x}_n) + \kappa)(1 - \texttt{SOFT}_c(F(\mathbf{x}_n))) \right] \tag{6}$$

where κ is a dataset specific constant, β is the learning rate, $\texttt{SOFT}_c$ is the softmax output for class c, and $V_{c,h,l}(\mathbf{x}_j) = \sum_{p \in \mathcal{P}_{\mathbf{v}_{c,l}}(\mathbf{x}_j)} \texttt{ReLU}(\langle \Theta_{c,h}, \mathbf{x}_j^p \rangle a^p)$ with $\mathcal{P}_{\mathbf{v}_{c,l}}(\mathbf{x}_j)$ being the collection of patches containing feature $\mathbf{v}_{c,l}$ in $\mathbf{x}_j$. The set $\mathcal{U}$ is an uncertainty set that assigns a weight to each data sample based on it loss. In particular, the uncertainty set under DRO is given as in (2) *and we further define the uncertainty set under ERM: $\mathcal{U}^{ERM} := \left\{ \mathbf{z} \in \mathbb{R}^N : z_n = \frac{1}{N}; \forall n \in [1, N] \right\}$. Learning via the robust loss in* (1) *leads to a stronger correlation between the network weights $\Theta_{c,h}$ and the single-view data feature $\mathbf{v}_{c,l}$:*

$$\{\langle \Theta_{c,h}^t, \mathbf{v}_{c,l} \rangle\}_{Robust} > \{\langle \Theta_{c,h}^t, \mathbf{v}_{c,l} \rangle\}_{ERM}; \forall t > 0 \tag{7}$$

Remark. The robust loss $\mathcal{L}^{\text{Robust}}$ forces the model to learn from the single-view samples (according to the loss) by assigning a higher weight. As a result, the network weights will be adjusted to increase the correlation with the single-view data features $\mathbf{v}_{c,l}$ due to Lemma 1. In contrast, for standard ERM, weight is uniformly assigned to all samples. Due to the sparse single-view data features (which also makes them more difficult to learn from, leading to a larger loss), the model does not grow sufficient correlation with $\mathbf{v}_{c,l}$. In this case, the ERM model instead learns to memorize some noisy feature $\mathbf{v}'$ introduced through certain spurious correlations. For a testing data sample, the ERM model may confidently assign it to an incorrect class k according to the noise feature $\mathbf{v}'$. In the theorem below, we show how the robust training proces can effectively lower the confidence of incorrect predictions, leading to an improved calibration performance.

Theorem 2. *Given a new testing sample $\mathbf{x} \in \mathcal{D}_S^N$ containing $\mathbf{v}_{c,l}$ as the main feature and a dominant noise feature $\mathbf{v}'$ that is learned due to memorization, we have*

$$\{\texttt{SOFT}_k(\mathbf{x})\}_{Robust} < \{\texttt{SOFT}_k(\mathbf{x})\}_{ERM} \tag{8}$$

where $\mathbf{v}'$ is assumed to be a main feature characterizing class k.

Remark. For ERM, due to the impact of the dominate noise feature $\mathbf{v}'$, it assigns a large probability to class k since $\mathbf{v}'$ is one of its major features, leading to high confidence for an incorrect prediction. In contrast, the robust learning process allows the model to learn a stronger correlation with the main feature $\mathbf{v}_{c,l}$ as shown in Lemma 1. Thus, the model is less impacted by the noise feature $\mathbf{v}'$, resulting in reduced confidence in predicting the wrong class k. Such a key property guarantees an improved calibration performance, which is clearly verified by our empirical evaluation. It is also worth noting that Theorem 2 does not necessarily lead to better classification accuracy. This is because (8) only ensures that that the false confidence is lower than an ERM model, but there is no guarantee that $\{\texttt{SOFT}_k(\mathbf{x})\}_{Robust} < \{\texttt{SOFT}_c(\mathbf{x})\}_{Robust}$. It should be noted that our DRE framework ensures diverse sparse sub-network focusing on different single-view data samples from different classes. As such, an ensemble of those diverse sparse subnetworks provides maximum coverage of all features (even the weaker one) and therefore can ultimately improve the calibration performance. The detailed proofs are provided in the Appendix.

4 Experiments

We perform extensive experimentation to evaluate the distributionally robust ensemble of sparse sub-networks. Specifically, we test the ability of our proposed technique in terms of calibration and classification accuracy. For this, we consider three settings: (a) general classification, (b) out-of-distribution setting where we have in-domain data but with different distributions, and (c) open-set detection, where we have unknown samples from new domains.

4.1 Experimental Settings

Dataset description. For the general classification setting, we consider three real-world datasets: Cifar10, Cifar100 [12], and TinyImageNet [14]. For the out-of-distribution setting, we consider the corrupted version of the Cifar10 and Cifar100 datasets which are named Cifar10-C and Cifar100-C [10]. It should be noted that in this setting, we train all models in clean dataset and perform testing in the corrupted datasets. For open-set detection, we use the SVHN dataset [19] as the open-set dataset and Cifar10 and Cifar100 as the close-set data. A more detailed description of each dataset is presented in the Appendix.

Evaluation metrics. To assess the model performance in the first two settings, we report the classification accuracy ($\mathcal{ACC}$) along with the Expected Calibration Error ($\mathcal{ECE}$). In the case of open-set detection, we report open-set detection for different confidence thresholds.

Implementation details. In all experiments, we use a family of ResNet architectures with two density levels: 9% and 15%. To construct an ensemble, we learn 3 sparse sub-networks each with a density of 3% for the total of 9% density and that of 5% density for the total of density 15%. All experiments are conducted with the 200 total epochs with an initial learning rate of 0.1 and a cosine scheduler function to decay the learning rate over time. The last-epoch model is taken for all analyses. For the training loss, we use the EP-loss in our DRO ensemble that optimizes the scores for each weight and finally selects the sub-network from the initialized dense network for the final prediction. The selection is performed based on the optimized scores. More detailed information about the training process and hyperparameter settings can be found in the Appendix.

4.2 Performance Comparison

In our comparison study, we include baselines that are relevant to our technique and therefore we primarily focus on the LTH-based techniques. Specifically, we include the initial lottery ticket hypothesis (LTH) [6] that iteratively performs pruning from a dense network until the randomly initialized sub-network with a given density is reached. Once the sub-network is found, the model trains the sub-network using the training dataset. Similarly, we also include L1 pruning [16]. We also include three approaches CigL [15], Sup-ticket [30], DST Ensemble [17] which are based on the pruning and regrowing sparse network training strategies. From Venkatesh et al. [28] we consider MixUp strategy as a comparison baseline as it does not require multi-step forward passes. A dense network is also included as a reference (denoted as *Dense*†). Furthermore, we report the performance obtained using the EP algorithm [23] on a single model with a given density. Finally, we also include the deep ensemble technique (*i.e.,* Sparse Network Ensemble (SNE), where each base model is randomly initialized and independently trained. The approaches that require pre-training of a dense network are categorized under the *Dense Pre-training* category. Those performing sparse network training but actually updating the network parameters are grouped as *Sparse Training*. It should be noted that sparse training techniques still require iterative pruning and regrowing. Finally, techniques

Table 1: Accuracy and ECE performance with 9% density for Cifar10 and Cifar100.

Training Type	Approach	Cifar10				Cifar100			
		ResNet50		ResNet101		ResNet101		ResNet152	
		$\mathcal{ACC}$	$\mathcal{ECE}$	$\mathcal{ACC}$	$\mathcal{ECE}$	$\mathcal{ACC}$	$\mathcal{ECE}$	$\mathcal{ACC}$	$\mathcal{ECE}$
	Dense†	94.82	5.87	95.12	5.99	76.40	16.89	77.97	16.73
Dense Pre-training	*L1 Pruning [16]*	93.45	5.31	93.67	6.14	75.11	15.89	75.12	16.24
	LTH [6]	92.65	3.68	92.87	6.02	74.09	15.45	74.41	16.12
	DLTH [2]	93.27	5.87	95.12	7.09	77.29	16.64	77.86	17.26
	Mixup [28]	92.86	3.68	93.06	6.01	74.15	15.41	74.28	16.05
Sparse Training	*CigL [15]*	92.39	5.06	93.41	4.60	76.40	9.30	76.46	9.91
	DST Ensemble [17]	88.87	2.02	84.93	0.8	63.57	7.23	63.22	6.18
	Sup-ticket [30]	94.52	3.30	95.04	3.10	78.28	10.20	78.60	10.50
Mask Training	*AdaBoost*	93.12	5.13	94.15	5.46	75.15	22.96	75.89	24.54
	EP [23]	94.20	3.97	94.35	4.03	75.05	14.62	75.68	14.41
	SNE	94.70	2.51	94.48	3.51	75.69	9.02	75.22	10.89
	DRE (Ours)	94.60	**0.7**	94.28	**0.7**	74.68	**1.20**	74.37	**2.09**

Table 2: Accuracy and ECE on TinyImageNet.

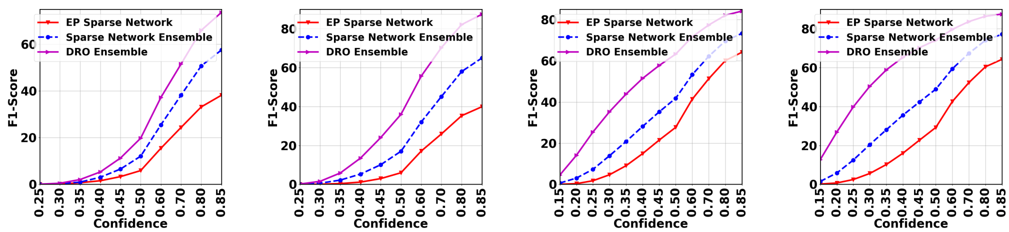
Training Type	Approach	$\mathcal{K} = 9\%$				$\mathcal{K} = 15\%$			
		ResNet101		WideResNet101		ResNet101		WideResNet101	
		$\mathcal{ACC}$	$\mathcal{ECE}$	$\mathcal{ACC}$	$\mathcal{ECE}$	$\mathcal{ACC}$	$\mathcal{ECE}$	$\mathcal{ACC}$	$\mathcal{ECE}$
	Dense†	71.28	15.58	72.57	16.96	71.28	15.58	72.57	16.96
Dense Pre-training	*L1 Pruning [16]*	68.85	14.72	69.78	16.38	70.24	14.24	70.98	15.36
	LTH [6]	69.23	13.97	69.13	15.34	70.16	13.63	70.25	14.24
	DLTH [2]	70.12	16.15	71.36	18.35	71.68	15.88	72.97	17.21
	Mixup [28]	69.34	14.24	69.25	15.59	70.28	14.31	70.39	14.57
Mask Training	*AdaBoost*	69.52	17.23	68.66	19.46	70.12	16.57	70.24	18.35
	EP [23]	69.88	10.78	71.57	9.82	70.46	11.99	70.71	12.41
	SNE	71.28	4.64	73.32	5.48	72.20	6.57	74.56	6.55
	DRE (Ours)	71.68	**3.48**	74.04	**2.82**	72.00	**1.52**	73.72	**1.08**

Table 3: Accuracy and ECE performance on out-of-distribution datasets.

Training Type	Approach	Cifar10				Cifar100			
		ResNet50		ResNet101		ResNet101		ResNet152	
		$\mathcal{ACC}$	$\mathcal{ECE}$	$\mathcal{ACC}$	$\mathcal{ECE}$	$\mathcal{ACC}$	$\mathcal{ECE}$	$\mathcal{ACC}$	$\mathcal{ECE}$
	Dense†	79.65	19.63	79.65	19.63	54.75	35.32	54.75	35.32
Dense Pre-training	*L1 Pruning [16]*	77.34	17.95	76.39	17.89	52.06	31.45	51.67	30.98
	LTH [6]	75.85	17.88	76.15	17.62	50.79	31.23	51.35	30.56
	DLTH [2]	79.67	21.74	80.12	20.31	54.82	37.55	55.12	35.74
	Mixup [28]	76.35	17.74	76.88	17.55	51.36	31.12	51.92	30.35
Sparse Training	*CigL [15]*	70.80	21.04	69.84	21.42	49.42	25.86	51.49	24.13
	Sup-ticket [30]	72.89	17.80	73.01	18.82	48.80	24.99	48.81	25.62
Mask Training	*AdaBoost*	75.94	22.96	74.55	21.46	51.36	38.45	51.25	38.34
	EP [23]	77.58	17.82	77.73	17.46	52.18	30.60	52.14	29.48
	SNE	78.93	15.73	78.61	15.56	54.74	24.22	54.00	20.54
	DRE (Ours)	78.57	**10.92**	78.00	**10.19**	54.11	**14.28**	53.21	**8.13**

that attempt to search the best initialized sparse sub-network through mask update (*e.g.,* EP) are grouped as *Mask Training*.

General classification setting. In this setting, we consider clean Cifar10, Cifar100, and TinyImageNet datasets. Tables 1, 2, and 10 (in the Appendix) show the accuracy and calibration error for different models with density 9% and 15%. It should be noted that for the TinyImageNet dataset, we could not run the Sparse Training techniques due to the computation issue (*i.e.,* memory overflow). This may be because sparse training techniques require maintaining additional parameters for the pruning and regrowing strategy. In the Appendix, we have made a comparison of the proposed DRE with those baselines on a lower architecture size. There are three key observations we can infer from the experimental results. First, sparse networks are able to maintain or improve the generalization performance (in terms of accuracy) with better calibration, which can be seen by comparing dense network performance with the edge-popup algorithm. Second, the ensemble in general helps to further lower the calibration error (lower the better). For example, in all datasets, standard ensemble



(a) CIFAR10 ($\mathcal{K}$ = 15%) (b) CIFAR10 ($\mathcal{K}$ = 9%) (c) CIFAR100 ($\mathcal{K}$ = 15%) (d) CIFAR100 ($\mathcal{K}$ = 9%)

Figure 3: Open-set detection performance on different confidence thresholds.

(SNE) consistently improves the EP model. Finally, the proposed DRE significantly improves the calibration performance by diversifying base learners and allow each sparse sub-network to focus on different parts of the training data. The strong calibration performance provides clear empirical evidence to justify our theoretical results.

Out-of-distribution classification setting. In this setting, we assess the effectiveness of the proposed techniques on out-of-distribution samples. Specifically, [10] provide the Cifar10-C and Cifar100-C validation datasets which are different than that of the original clean datasets. They apply different corruptions (such as blurring noise, and compression) to shift the distribution of the datasets. We assess those corrupted datasets using the models trained using the clean dataset. Table 3 shows the performance using different architectures. In this setting, we have not included DST Ensemble, because: (a) its accuracy is far below the SOTA performance, and (b) same training mechanism as that of the Sup-ticket, whose performance is reported. As shown, the proposed DRE provides much better calibration performance even with the out of distribution datasets.

Open-set detection setting. In this setting, we demonstrate the ability of our proposed DRO ensemble in detecting open-set samples. For this, we use the SVHN dataset as an open-set dataset. Specifically, if we have a better calibration, we would be able to better differentiate the open-set samples based on the confidence threshold. For this, we randomly consider 20% of the total testing in-distribution dataset as the open-set samples from the SVHN dataset. The reason for only choosing a subset of the dataset is to imitate the practical scenario where we have very few open-set samples compared to the close-set samples. We treat the open-set samples as the positive and in-distribution (close-set) ones as the negative. Since this is a binary detection problem, we compute the F-score [8] at various thresholds, which considers both precision and recall. Figure 3 shows the performance for the proposed technique along with comparative baselines. As shown, our proposed DRE (refereed as DRO Ensemble) always stays on the top for various confidence thresholds which demonstrates that strong calibration performance can benefit DRE for open-set detection as compared to other baselines.

4.3 Additional Results, Ablation Study, and Qualitative Analysis

Limited by space, we have reported additional results in the Appendix. Specifically, we compare the proposed DRE with other standard calibration techniques commonly used in dense networks. In addition, we have performed an ablation study to investigate the impact of parameter η and different backbones (*i.e.,* ViT and WideResNet). We present a qualitative analysis to further justify the effectiveness of our proposed technique. Finally, we report the parameter size and inference speed (FLOPS) of DRE and compare it with existing baselines.

5 Conclusion

In this paper, we proposed a novel DRO framework, called DRE, that achieves an ensemble of lottery tickets towards calibrated network sparsification. Specifically, with the guidance of uncertainty sets under the DRO framework, the proposed DRE aims to learn multiple diverse and complementary sparse sub-networks (tickets) where uncertainty sets encourage tickets to gradually capture different data distributions from easy to hard and naturally complement each other. We have theoretically justified the strong calibration performance by demonstrating how the proposed robust training process guarantees to lower the confidence of incorrect predictions. The extensive evaluation shows that the proposed DRE leads to significant calibration improvement without sacrificing the accuracy and burdening inference cost. Furthermore, experiments on OOD and Open-set datasets show its effectiveness in terms of generalization and novelty detection capability, respectively.

References

[1] Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[2] Yue Bai, Huan Wang, ZHIQIANG TAO, Kunpeng Li, and Yun Fu. Dual lottery ticket hypothesis. In *International Conference on Learning Representations*, 2022.

[3] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars, 2016.

[4] Tianlong Chen, Zhenyu Zhang, Jun Wu, Randy Huang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Can you win everything with a lottery ticket? *Transactions on Machine Learning Research*, 2022.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[6] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. 2018.

[7] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning, 2015.

[8] Cyril Goutte and Eric Gaussier. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In David E. Losada and Juan M. Fernández-Luna, editors, *Advances in Information Retrieval*, pages 345–359, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[9] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1321–1330. JMLR.org, 2017.

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

[11] Xiaoqian Jiang, Melanie Osl, Jihoon Kim, and Lucila Ohno-Machado. Calibrating predictive model estimates to support personalized medicine. *Journal of the American Medical Informatics Association : JAMIA*, 19:263 – 274, 2011.

[12] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[13] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6405–6416, Red Hook, NY, USA, 2017. Curran Associates Inc.

[14] Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge. 2015.

[15] Bowen Lei, Ruqi Zhang, Dongkuan Xu, and Bani Mallick. Calibrating the rigged lottery: Making all tickets reliable. In *The Eleventh International Conference on Learning Representations*, 2023.

[16] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets, 2016.

[17] Shiwei Liu, Tianlong Chen, Zahra Atashgahi, Xiaohan Chen, Ghada Sokar, Elena Mocanu, Mykola Pechenizkiy, Zhangyang Wang, and Decebal Constantin Mocanu. Deep ensembling with no overhead for either training or testing: The all-round blessings of dynamic sparsity, 2022.

[18] Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2901–2907. AAAI Press, 2015.

[19] Yuval Netzer, Tao Wang, Adam Coates, A. Bissacco, Bo Wu, and A. Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[20] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift, 2019.

[21] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions, 2017.

[22] John Platt and Nikos Karampatziakis. Probabilistic outputs for svms and comparisons to regularized likelihood methods. 2007.

[23] Vivek Ramanujan, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari. What's hidden in a randomly weighted neural network?, 2019.

[24] Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations, 2020.

[25] Robert E Schapire. Explaining adaboost. In *Empirical inference*, pages 37–52. Springer, 2013.

[26] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty, 2018.

[27] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision, 2015.

[28] Bindya Venkatesh, Jayaraman J. Thiagarajan, Kowshik Thopalli, and Prasanna Sattigeri. Calibrate and prune: Improving reliability of lottery tickets through prediction calibration, 2020.

[29] Andrew G Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4697–4708, 2020.

[30] Lu Yin, Vlado Menkovski, Meng Fang, Tianjin Huang, Yulong Pei, Mykola Pechenizkiy, Decebal Constantin Mocanu, and Shiwei Liu. Superposing many tickets into one: A performance booster for sparse neural network training, 2022.

[31] Jize Zhang, Bhavya Kailkhura, and T. Yong-Jin Han. Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning, 2020.

[32] Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. Deconstructing lottery tickets: Zeros, signs, and the supermask, 2019.