Improving Classifier Confidence using Lossy Label-Invariant Transformations

Sooyong Jang
PRECISE Center
University of Pennsylvania

Insup Lee
PRECISE Center
University of Pennsylvania

James Weimer
PRECISE Center
University of Pennsylvania

Abstract

Providing reliable model uncertainty estimates is imperative to enabling robust decision making by autonomous agents and humans alike. While recently there have been significant advances in confidence calibration for trained models, examples with poor calibration persist in most calibrated models. Consequently, multiple techniques have been proposed that leverage label-invariant transformations of the input (*i.e.,* an input manifold) to improve worst-case confidence calibration. However, manifold-based confidence calibration techniques generally do not scale and/or require expensive retraining when applied to models with large input spaces (*e.g.,* ImageNet). In this paper, we present the recursive lossy label-invariant calibration (ReCal) technique that leverages label-invariant transformations of the input that induce a loss of discriminatory information to recursively group (and calibrate) inputs – without requiring model retraining. We show that ReCal outperforms other calibration methods on multiple datasets, especially, on large-scale datasets such as ImageNet.

1 Introduction

Despite the success of machine learning predictions in various applications including image classifications (He et al., 2016; Zagoruyko and Komodakis, 2016; Xie et al., 2017), speech recognition (Graves et al., 2013; Wang et al., 2019), games (Mnih et al., 2013; Silver et al., 2017), and medical research (Rajpurkar et al., 2018), estimating prediction confidence has a different story. As observed in Guo et al. (2017), many modern neural networks are miscalibrated, *i.e.,* they are over-confident in their predictions. As machine learning expands to safety-critical applications such as self-driving cars, autonomous pilots, and autonomous medical systems, accurately estimating confidence becomes imperative for robust decision making.

Consequently, various approaches have been introduced to address the problem of estimating confidence. Bayesian techniques (Gal and Ghahramani, 2016; Zhang et al., 2017; Khan et al., 2018; Chang et al., 2019) provide a means of computing the posterior distribution of models for estimating confidence, but suffer from computational limitations. Also proposed are techniques that change the original model estimates (Tran et al., 2019; Kumar et al., 2018; Seo et al., 2019), but these techniques have the disadvantage that they require re-training the model and do not guarantee the accuracy of the original model. Lastly, there have been many post-hoc approaches proposed that learn a model mapping uncalibrated confidence to calibrated confidence on a comparatively small validation set *e.g.,* temperature scaling, vector scaling (Guo et al., 2017), using spline (Gupta et al., 2020), MS-ODIR and Dir-ODIR (Kull et al., 2019), mix-n-match (Zhang et al., 2020), GPcalib (Wenger et al., 2020), and intra-order preserving functions (Rahimi et al., 2020). While these techniques provide improved average confidence calibration, poorly calibrated examples remain.

To address this issue, techniques that utilize redundancy in the example space have been proposed (Bahat and Shakhnarovich, 2020; Thulasidasan et al., 2019; Patel et al., 2019). The premise behind these techniques is that utilizing additional information on the current sample, its confidence calibration can be improved. Most of these techniques augment the training dataset with examples on the same manifold and re-train a model on the augmented dataset (Thulasidasan et al., 2019; Patel et al., 2019). While other techniques (Bahat and Shakhnarovich, 2020), avoid retraining by assuming there exist well calibrated examples within the manifold and perform filtering of a sampling of the manifold confidences. While evaluating the confidence of an individual sample remains a challenge – since most benchmark datasets do not contain confidence labels (only class labels) – these techniques do generally show marked improvement in expected confidence calibration consistent with a reduced number of poorly calibrated samples. However, manifold-based techniques have severe shortcomings

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

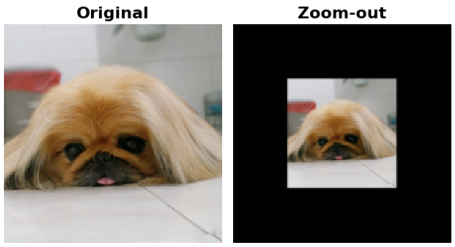

Figure 1: Example of Original and Transformed Image

when applied to datasets with large input spaces. Augmenting the training dataset and retraining a model scales poorly as the input space (and manifold dimensionality) increases. Similarly, filtering sampled confidences assumes that the original classifier is calibrated for a majority of the manifold, which becomes less likely as manifold dimensionality increases. Consequently, manifold-based calibration of models for large input spaces remains a challenge.

In this work, we present *Recursive Lossy Label-Invariant Calibration* (or ReCal) as a scalable manifold-based post-hoc confidence calibration algorithm that maintains the accuracy of the original classifier and scales to large datasets (*e.g.* ImageNet). To overcome the scalability issues of other manifold-based techniques, we only consider label-invariant transformations that are expected to result in a decreased confidence due to discriminatory information loss – *i.e.,* lossy label-invariant transformations. For example, consider zooming out an image of a dog with the scale factor of 0.5x as shown in Figure 1. After the transformation, the image still contains the dog, but the dog becomes smaller and harder to recognize. Therefore, we should be able to identify the dog but with less confidence. Considering this intuition in the context of estimating confidence, a (well-calibrated) classifier should return the same prediction with smaller confidence after applying a lossy label-invariant transformation. Likewise, if we group examples based on the prediction and confidence change after such transformations, we expect that the examples in the same group will have similar properties respect to the classifier and confidence estimation. In other words, examples in each group require a similar amount of adjustment, which may be different than the adjustment needed for examples in other groups. This intuition – lossy label-invariant grouping – forms the premise of ReCal, and is discussed in detail in Section 4.

Leveraging group-wise calibration, we propose ReCal as a scalable post-hoc calibration algorithm in Section 5. Specifically, the proposed algorithm recursively leverages lossy label-invariant transformations to re-group images and perform group-wise calibration. Different from other approaches that aim to retrain a model on the augmented training set (Patel et al., 2019; Thulasidasan et al., 2019), our proposed algorithm does not change the predictions and thus retains the original prediction accuracy while adjusting the confidence of the predictions.

We demonstrate the scalability and performance of the proposed algorithm by applying it to ImageNet, and also compare ReCal with other calibration algorithms on CIFAR10/100, ImageNet in Section 6. On multiple models *e.g.,* LeNet5, DenseNet, ResNet, ResNet SD, and Wide ResNet, on the datasets, we compare Expected Calibration Error (ECE) (Naeini et al., 2015), Brier score (Brier, 1950) and time for learning a calibration map. On the large scale image dataset, ImageNet, ReCal can be applied to the dataset in terms of time, and it outperforms other calibration algorithms such as temperature scaling, vector scaling (Guo et al., 2017), MS-ODIR, Dir-ODIR (Kull et al., 2019) on DenseNet161 and ResNet152 models. Besides ImageNet, ReCal shows the best performance or the second-best performance for seven of ten models on CIFAR10/100.

The contributions of this paper are summarized as follows:

- introducing lossy label-invariant grouping and empirically demonstrating that each group needs different calibration;
- presenting ReCal, a scalable post-hoc calibration algorithm based on lossy label-invariant transformation, which can be applied to a large-scale datasets;
- evaluating ReCal in comparison to other publicly released post-hoc calibration algorithms using multiple datasets and models.

The remainder of this paper is structured as follows. In the next section, we present the related work on confidence calibration. In Section 3, we present the problem statement considered herein. Section 4 describes lossy label-invariant grouping and its effectiveness. We then propose ReCal in Section 5, present the experimental results in Section 6, and conclusions in Section 7.

2 Related Work

While a complete review of all confidence calibration techniques is beyond the scope of this work, in this section we selectively review those techniques most related to the proposed approach. In the following, we consider confidence calibration techniques leveraging Bayesian uncertainty estimation, calibration via re-training, post-hoc calibration maps, and manifold-based calibration.

Bayesian uncertainty estimation. One approach to confidence calibration is to provide uncertainty estimation with Bayesian framework (Gal and Ghahramani, 2016; Zhang et al., 2017; Khan et al., 2018; Chang et al., 2019). While Bayesian techniques can provide very accurate calibration, they suffer from computational limitations associated with estimating the posterior distribution used for uncertainty estimation.

Calibration via re-training. Another type of approach targets training a well-calibrated classifier (Kumar et al., 2018; Lakshminarayanan et al., 2017; Seo et al., 2019; Tran et al., 2019; Müller et al., 2019). A potential pitfall of calibration via re-training is that the accuracy of the prediction can change. Moreover, many of these approaches require training sophisticated networks on large training datasets, which may consume significant time and computational resources.

Post-hoc calibration maps. Post-hoc methods address the calibration problem without requiring model retraining. These approaches employ binning methods such as Histogram Binning (Zadrozny and Elkan, 2001), Bayesian Binning into Quantiles (Naeini et al., 2015), Mutual Information Maximization-based Binning (Patel et al., 2020) or train a function mapping from original confidence to calibrated one on validation data which is smaller compared to the training data. For training a mapping function, several techniques have been proposed (Platt et al., 1999; Zadrozny and Elkan, 2002; Guo et al., 2017; Rahimi et al., 2020; Gupta et al., 2020; Kull et al., 2019; Zhang et al., 2020; Wenger et al., 2020). Most notably, Guo et al. (2017), introduces temperature scaling which transforms original logits to calibrated logits with a single parameter. Besides temperature scaling, intra-order preserving function (Rahimi et al., 2020), Dirichlet calibration with ODIR regularization (Kull et al., 2019), splines (Gupta et al., 2020), latent Gaussian function (GPcalib) (Wenger et al., 2020), and ETS, IRM, IROvA-TS (Zhang et al., 2020) have been proposed. Depending on the mapping function, some of the approaches such as temperature scaling, intra-order preserving function, splines, ETS and IRM preserve the accuracy, while the others like matrix scaling, vector scaling, IROvA-TS, GPcalib and Dirichlet calibration do not preserve the original model accuracy.

Manifold-based calibration. Several manifold-based confidence calibration have been proposed (Bahat and Shakhnarovich, 2020; Thulasidasan et al., 2019; Patel et al., 2019; Lee et al., 2017; Verma et al., 2019). Bahat and Shakhnarovich (2020) augments test data using transformations to calibrate confidence, while Thulasidasan et al. (2019) and Patel et al. (2019) augment data by interpolating existing data and using an auto-encoder based model, respectively. Other techniques augment the training data with samples from the manifold and retrain the model (Lee et al., 2017; Verma et al., 2019). Manifold-based algorithms can improve worst-case calibration errors as shown by their ability to address over-confident prediction on out-of-distribution samples. However, they generally suffer from scalability issues as discussed in Section 1.

3 Problem Statement

In this paper, we aim to develop a post-hoc calibration algorithm which addresses the worst-case confidence error that does not change accuracy on a multi-class classification task. Consider a multi-class classification task on data, $D = \{(x_n, y_n)\}^{N_D} \sim \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is an input space and $\mathcal{Y}$ is a label set, $\{1, 2, \ldots, K\}$. Let $f : \mathcal{X} \to \mathbb{R}^K$ denote a multi-class classifier. A neural network classifier typically has a softmax output layer as a final layer, which returns a vector p for the given input x. Here, $p = f(x) = \{p_1, p_2, \ldots, p_K\}$. Each p_i is the estimated probability of a label i, and the classifier chooses the label whose probability is the maximum. Consequently, the prediction $\hat{y} = \text{argmax}_i\{p\}$ has confidence $\hat{p} = \max_i\{p\}$.

A classifier f is *calibrated* if confidence is equal to accuracy given the confidence. More formally,

$$\mathbf{P}[y = k | p_k = p'] = p' \tag{1}$$

where, $p = f(x), k = \text{argmax}_i\{p\}$ for all $(x, y) \in D$ and for all $p' \in [0, 1]$. Here, the difference between the both sides is estimated by *Expected Calibration Error (ECE)* (Naeini et al., 2015) which is the weighted average of the differences over bins. ECE is computed by first splitting the confidence range with equal size bins, and calculating the difference between average confidence and average accuracy of each bins, and finally, computing an average of those differences weighted by the number of samples in each bins. More formally,

$$\text{ECE} = \sum_{i=1}^{M} \frac{|B_i|}{|D|} \left(|accuracy(B_i) - confidence(B_i)|\right) \tag{2}$$

where, M is the number of bins, $B_1, \ldots, B_M$ are bins which equally divides the interval [0, 1], and $accuracy(B_i)$ is the average accuracy of examples in bin B_i, and $confidence(B_i)$ is the average confidence of examples in bin B_i.

Ideally, we would like to minimize the worst-case confidence error, however, this is impossible to quantify with the current datasets that lack ground truth calibration values for the labels. As a surrogate, and consistent with other works in the literature, we rather aim to minimize ECE. Therefore, we would like to learn a calibration map which transforms the original confidence (or logits) to calibrated one which minimize ECE without affecting original accuracy.

4 Lossy Label-Invariant Grouping

It is reasonable to assume that different examples may need different level of adjustment for the calibration, *i.e.,* some examples require more adjustment than others. Consequently, we would like to group inputs based on some measure of adjustment needed so that we can apply different level of adjustment to each group. In other words, we would like to apply more adjustment when predictions are very mis-calibrated, and adjust less when predictions near calibration.

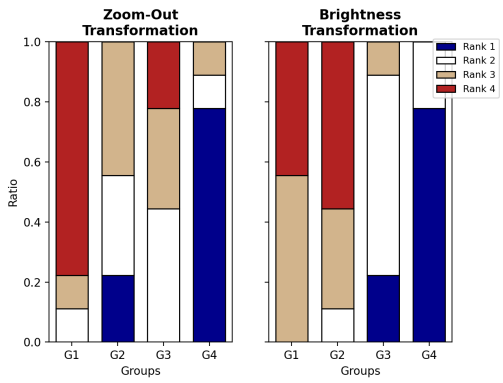

Figure 2: Rank Distribution of Each Group with Two Transformations. Each Bar Represents Rank Distribution of Each Group over Different Transformation Parameters.

We utilize a subset of transformations that do not change the label called *label-invariant transformations*. Specifically, we choose label-invariant transformations which induce a loss in discriminatory information – *i.e.,* lossy label-invariant transformations. As an example, consider an image classification task. The zoom-out transformation and brightness transformation are the examples of lossy label-invariant transformations. These two transformation do not change label, but reduce discriminatory information by making an image smaller or darker. Therefore, after such transformations, a (well-calibrated) classifier should not change its prediction but should become less confident on its prediction.

Our approach, as described in Algorithm 1, begins by applying a lossy label-invariant transformation to the inputs, and group based on the observed prediction and confidence changes after the transformation. There can be two possible outcomes to each observation, *i.e.,* prediction change vs. not change, and confidence increase vs. not increase, and in total there can be four possible combinations as shown in Table 1. We perform *lossy label-invariant grouping* by comparing the prediction and confidence of the transformed input with the original input, and group based on the comparison result. More formally, group number k for an input is

$$k = 2 \times \mathbb{1}_{(\hat{y}=\hat{y}_t)} + \mathbb{1}_{(\hat{p}\geq\hat{p}_t)} + 1 \tag{3}$$

where, $\hat{y}, \hat{p}$ are the prediction and confidence for the original input, $\hat{y}_t, \hat{p}_t$ are the prediction and confidence for the transformed input, and $\mathbb{1}_{(\cdot)}$ is 1 if $(\cdot)$ is true, and 0, otherwise.

To demonstrate the effectiveness of our Lossy Label-Invariant Grouping Algorithm we consider an image classification task using ResNet152 model on ImageNet. We choose two image transformations, zoom-out and brightness. These two transformations have one parameter which determines how much transformation will be applied. A

Algorithm 1 Lossy Label-Invariant Grouping

procedure GRP_INPUT(z, z_t)
  Input: z : $(N_v \times K)$ Original inputs logits; z_t : $(N_v \times K)$ Transformed inputs confidence logits
  $\hat{y} \leftarrow$ argmax(p, axis=1)
  $\hat{y}_t \leftarrow$ argmax(p_t, axis=1)
  $\hat{p} \leftarrow$ softmax(z, axis=1)$^{\hat{y}}$
  $\hat{p}_t \leftarrow$ softmax(z, axis=1)$^{\hat{y}}$
  $g\_1\_idx = (\hat{y} \neq \hat{y}_t) \wedge (\hat{p}_t > \hat{p})$
  $g\_2\_idx = (\hat{y} \neq \hat{y}_t) \wedge (\hat{p}_t \leq \hat{p})$
  $g\_3\_idx = (\hat{y} = \hat{y}_t) \wedge (\hat{p}_t > \hat{p})$
  $g\_4\_idx = (\hat{y} = \hat{y}_t) \wedge (\hat{p}_t \leq \hat{p})$
  return $g\_1\_idx, g\_2\_idx, g\_3\_idx, g\_4\_idx$
end procedure

Table 1: Grouping Inputs Based On Prediction and Confidence Change. $\hat{y}, \hat{p}$ Are the Prediction and Confidence for Original Input, and $\hat{y}_t, \hat{p}_t$ Are the Prediction and Confidence for Transformed Input.

	$\hat{p} < \hat{p}_t$	$\hat{p} \geq \hat{p}_t$
$\hat{y} \neq \hat{y}$	Group 1	Group 2
$\hat{y} = \hat{y}$	Group 3	Group 4

zoom-out transformation with smaller parameter value will return the smaller image and a brightness transformation with smaller parameter value will yield the darker image.

We randomly select parameter values between 0.1 and 0.9, observe label prediction change and confidence change for validation images, and group images into four different groups as shown in Table 1. For each parameter, we compute and rank the ECE values among four groups, and draw the distribution of the ranks for each transformation as shown in Figure 2 . For the reference, ECE values and number of images of each group for the two transformations are displayed in the supplementary material.

In the figure, the x-axis is for the group index and the y-axis is for the proportion of transformation parameters which has the specific rank for the specific group. For example, in the left figure, G1 has about 80% for 'Rank 4', 10% for 'Rank 3', and 10% for 'Rank 2'. This means that for about 80% of all sampled parameters, Group 1 has the worst ECE among the four groups.

As shown in Figure 2 (Left), with zoom-out transformation, for about 80% of parameters, Group 4 which represents that prediction does not change and confidence does not increase, has the best ECE, *i.e.,* rank one. On the other hand, Group 1 which corresponds to the case that prediction changes and confidence increases shows the worst ECE, *i.e.,* rank four. The similar pattern appears on the brightness transformation as shown in Figure 2 (Right). With brightness transformation, Group 4 has the best ECE for about 80% of parameters, and Group 1 has the worst

ECE for about 50% of parameters. Group 2 and Group 3 show a little different pattern between two transformations. For zoom-out transformation, Group 2 has better rank than Group 3 with about 80% of each group is either rank 2 or 3. On the other hand, for brightness transformation, Group 2 has worse rank than Group 3. It is hard to decide which one requires more adjustment, but in general, these two groups should be calibrated differently.

These results empirically demonstrate that lossy label-invariant grouping partitions the inputs into groups that require different amounts of adjustment. Group 4 inputs which match our intuition tend to have the best ECE, *i.e.,* requires the least adjustment, while Group 1 inputs which opposite to our intuition show the worst ECE, *i.e.,* requires the most adjustment. Furthermore, input grouping differs depending on the transformation, as shown by the input distribution over groups for different transformations in Table 4 and 5 in supplementary material. Consequently, in the following section, we design an algorithm that utilizes different lossy label-invariant transformations at each iteration to perturb the groupings and perform recursive calibration.

5 Recursive Lossy Label-Invariant Calibration (ReCal)

As illustrated in Figure 3 and Algorithm 2, ReCal consists of 3 steps: initialization, iterative group-wise calibration, final calibration. In the following we describe each of these steps in detail. We conclude this section by analyzing the convergence of ReCal, presenting a runtime implementation of ReCal and a discussion of its limitations.

5.1 Initialization

To initialize ReCal, a transformation pool is prepared by sampling N transformations $\{t_1, \ldots, t_N\}$ from possible transformations. After a transformation pool is prepared, ReCal computes base logits of the original inputs and transformed inputs. The logits of the original inputs are obtained by feeding the original inputs to the original confidence estimator. For the logits of the transformed inputs, the original inputs are transformed by the sampled N transformations, and fed to the same original confidence estimator.

5.2 Iterative Group-Wise Calibration

The following three steps will be repeated up to the maximum iteration, L, or until the stopping condition is satisfied: (i) Transformation sampling; (ii) Lossy label-invariant grouping; (iii) Temperature scaling and logits update. This algorithm is described in Line 7 - 15 of Algorithm 2 and each step is detailed below.

Transformation sampling. At each iteration l, a transformation t^l is randomly sampled with replacement from a transformation pool $\{t_1, \ldots, t_N\}$.

Lossy label-invariant grouping. Once a transformation t^l is sampled, inputs in a validation set will be grouped using the lossy label-invariant grouping algorithm presented in Section 4.

Temperature scaling and logits update. For each lossy label-invariant group, temperature scaling (Guo et al., 2017) is applied, and temperature parameters, $\hat{\sigma}_1^l, \hat{\sigma}_2^l, \hat{\sigma}_3^l, \hat{\sigma}_4^l$, are generated – one corresponding to each group. Each group will have different number of inputs and overfitting can occur if the number of inputs is small. To safeguard against overfitting, we modify the temperature parameter based on the number of inputs in the group as shown in Equation 4.

$$\sigma_k^l = \left(1 - \frac{|G_k|}{|D_{val}|}\right) \times 1 + \frac{|G_k|}{|D_{val}|} \times \hat{\sigma}_k^l \tag{4}$$

where, $|G_k|$ is the number of inputs in group k and $|D_{val}|$ is the number of inputs in validation set. For example, if all the inputs belong to G_k, the temperature parameter from the temperature scaling will be used, and if there is no inputs in G_k, the temperature parameter is equal to 1, which means no calibration will be applied to G_k. With these modified temperature parameters, both original inputs logits and transformed inputs logits are computed. These temperature parameters and updated logits are stored for the test time and the later iterations, respectively.

Convergence Analysis Each iteration of ReCal aims to minimize the ECE within each of the four groups. Since $\sigma_i = 1$ is always a feasible solution for each group-wise calibration – corresponding to no change in calibration error – it follows that the ECE within each group is non-increasing. Further, the population ECE is also non-increasing since it is a weighted average of the group-wise ECE. Thus, the likelihood of satisfying the ReCal exit condition – which represents convergence of the ECE – increases with each iteration. While this assures an eventual exit, the rate of convergence is domain specific and depends on the sample data as well as the transformations employed.

5.3 Runtime Confidence Calculation using ReCal

After the calibration on validation set finished, inputs in test set can be calibrated as described in Algorithm 3. Because the transformation pool and a transformation at each iteration are already prepared in the calibration step, applying calibration step starts with computing base logits. After the base logits of original inputs and transformation inputs are ready, the iterative procedures will be repeated for L^* iterations, as determined in the calibration step.

Specifically, the iterative steps at runtime uses L^* sampled transformations and $4 \times L^*$ temperature parameters from the calibration step; at each iteration, one transformation is sampled and a temperature parameter is computed for each of the four groups. For each iteration, inputs are grouped

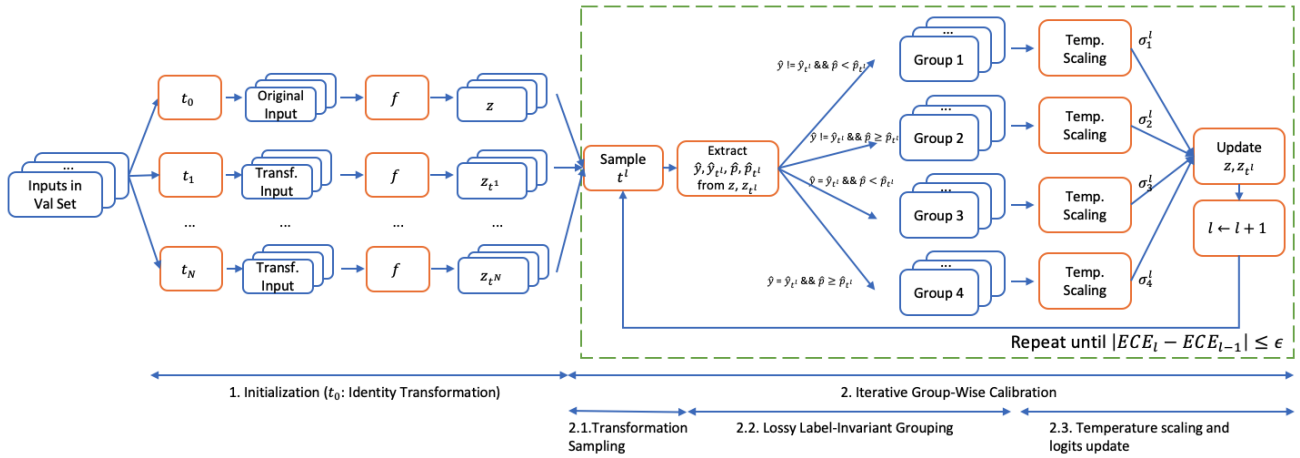

Figure 3: Illustration of Recursive Lossy Label-Invaraint Calibration (ReCal)

Algorithm 2 Recursive Lossy Label-Invariant Calibration (ReCal)

procedure RE_CAL(ts, X, y, N, L, δ)
  Input: ts: $(N_{allow} \times 1)$ Transformations specification; X: $(N_v \times p)$. Inputs in validation set; y:$(N_v \times 1)$. True labels; N:(1×1). Number of transformations; L:(1×1). Maximum iteration number; δ:(1×1). Stopping iterations threshold
  $\{t_1, \dots t_N\} \leftarrow$ Build a transformation pool based on ts
  $z \leftarrow$ Base logits for original inputs
  $z_{t_1}, \dots z_{t_N} \leftarrow$ Base logits for transformed inputs
  for $l = 1, 2, \dots, L$ **do**
    $t^l \leftarrow$ Randomly select a transformation from $\{t_1, \dots t_N\}$
    Group logits z and z_{t^l} using grp_img in Algorithm 1
    Apply temperature scaling to group 1 - 4 and obtain temperature parameters, $\hat{\sigma}_1^l, \hat{\sigma}_2^l, \hat{\sigma}_3^l, \hat{\sigma}_4^l$,
    Compute temperature parameters $\sigma_1^l, \sigma_2^l, \sigma_3^l, \sigma_4^l$ using Equation 4
    Calibrate logits for original inputs, z^1, z^2, z^3, z^4 and logits for transformed inputs, $z_{t^l}^1, z_{t^l}^2, z_{t^l}^3, z_{t^l}^4$
    Update logits for original inputs z using z^1, z^2, z^3, z^4, and logits for transformed inputs z_{t^l} using $z_{t^l}^1, z_{t^l}^2, z_{t^l}^3, z_{t^l}^4$
    if $|ECE_l - ECE_{l-1}| < \delta$ **then**
      return $\sigma_1^1, \sigma_2^1, \sigma_3^1, \sigma_4^1, \dots, \sigma_1^l, \sigma_2^l, \sigma_3^l, \sigma_4^l, t^1, \dots, t^l, l$
    end if
  end for
end procedure

Algorithm 3 Runtime Confidence Calculation using ReCal

procedure APPLY_CAL($X, \sigma_1^1, \sigma_2^1, \sigma_3^1, \sigma_4^1, \dots, \sigma_1^{L^*}, \sigma_2^{L^*}, \sigma_3^{L^*}, \sigma_4^{L^*}, t^1, \dots, t^{L^*}, L^*$)
  Input: X: $(N_{te} \times p)$. Test set inputs; $\sigma_1^1, \sigma_2^1, \sigma_3^1, \sigma_4^1, \dots, \sigma_1^{L^*}, \sigma_2^{L^*}, \sigma_3^{L^*}, \sigma_4^{L^*}$: $(4L^* \times 1)$. Temperature parameters ;$t^1, \dots, t^{L^*}$: $(L^* \times 1)$. Sampled transformation for each iteration; L^*:(1×1). Iteration number
  $z \leftarrow$ Base logits for original inputs
  $z_{t_1}, \dots z_{t_N} \leftarrow$ Base logits for transformed inputs
  for $l = 1, 2, \dots, l^*$ **do**
    Calibrate original inputs logits, z^1, z^2, z^3, z^4
    Calibrate transformed inputs logits, $z_{t^l}^1, z_{t^l}^2, z_{t^l}^3, z_{t^l}^4$
    Update original inputs logits z using z^1, z^2, z^3, z^4
    Update transformed inputs logits z_{t^l} using $z_{t^l}^1, z_{t^l}^2, z_{t^l}^3, z_{t^l}^4$
  end for
  return updated logits for original inputs
end procedure

using the sampled transformation, and confidences for the inputs are adjusted using the temperature parameters assigned to the group of each input.

5.4 Limitations of ReCal

ReCal has a few limitations. First, we have to have a label-invariant transformation which lose information, which is necessary for lossy label-invariant grouping step. However, it is not hard to find such transformations. For example, in image classifications, besides zoom-out transformation and brightness transformation used in this paper, image blurring / pixalization, lossy compression, and random pixel changes are other possible examples. Regarding classification on time-series data such as video and medical data, introducing missing data is a type of lossy label-invariant transformation since the act of losing a frame or occasional data sample doesn't change the state of the environment or patient.

Next, ReCal needs to consider N possible transformed inputs which may be inefficient memory-wise. However, what we mainly use is the logits not the inputs, since once we compute the base logits at the beginning, we do not use the inputs anymore. This logits are much smaller compared to the inputs because the logits size is equal to the number of classes and in classification task. For example, each ImageNet input data has a dimension of $224 \times 224 \times 3$ which takes about 588 KB, while the logits have a dimension of $1,000$, which is about 3.9 KB.

6 Experiments

We apply ReCal to multiple models on three datasets to compare its calibration performance and scalability. In detail, we train or obtain models for each dataset, and calibrate confidence using ReCal and other baselines. We then compare the calibration performance using two metrics and the time for learning a calibration map to evaluate the scalability. The details of datasets, model, baselines, metrics and results are described in the following subsections.

6.1 Experimental Setup

This subsection will explain the datasets, models, baselines, and evaluation metrics for the experiments. In detail, the first subsection briefly describes datasets and models used for each dataset. The next subsection is for describing what other calibration algorithms is used as baselines, and the final subsection illustrates the evaluation metrics.

Datasets and Models. We perform experiments on three datasets: CIFAR10, CIFAR100 (Krizhevsky et al., 2009), and ImageNet (Deng et al., 2009). For CIFAR10/100, we use DenseNet40 (Huang et al., 2017), LeNet5 (LeCun et al., 1998), ResNet110 (He et al., 2016), ResNet110 SD (Huang et al., 2016), and WRN-28-10 (Zagoruyko and Komodakis, 2016). For ImageNet, we use DenseNet161 (Huang et al., 2017) and ResNet152 (He et al., 2016). Complete details of the datasets and models are provided in the supplementary material.

Competing Approaches for Baseline Comparisons. We compare ReCal with various other calibration methods such as temperature scaling, vector scaling, MS-ODIR, Dir-ODIR. Among those methods, temperature scaling keeps the original accuracy, and other methods change the accuracy. We implement temperature scaling, and obtain codes for vector scaling, MS-ODIR, and Dir-ODIR from the paper's repository (Kull et al., 2019).

Evaluation Metrics. Our main goal is minimizing the worst-case confidence error, however, as described in Section 3, it is impossible to quantify due to the absence of available datasets with confidence estimates for the label. Instead, we aim to minimize ECE, and our main evaluation metric for the experiments is ECE. Besides ECE, we also compare approaches using Brier score (Brier, 1950), which considers accuracy as well. For completeness, definitions of ECE and Brier score are provided in the supplementary material. Lastly, for assessing the scalability, we compute the learning time of a calibration map.

6.2 Results

We analyze the results in terms of calibration performance and time for learning a calibration map. First, we compare the calibration performance in terms of ECE and brier score. ECE is for evaluating how well each algorithms calibrate confidence. Brier score is for the similar evaluation, but, this metric considers the prediction accuracy together. Second, we present the time for learning a calibration map so that we assess the scalability.

6.2.1 Calibration Performance

We display the calibration performance of various methods in Table 2 and 3. Table 2 and 3 display ECE and Brier score, and test error rates are shown in supplementary material. The values with bold and with underline represent the best and the second best result, respectively.

For ReCal, we show the three different transformation pools: (z, .1-.9, 20), (z, .5-.9, 10), (b, .1-.9, 20). The first parameter means the transformation type; z and b mean zoom-out transformation and brightness transformation, respectively. Next parameter represents the range of transformation parameters, the range is either from 0.1 to 0.9 or from 0.5 to 0.9. The last parameter corresponds to the number of transformation. We use 20 transformations when we have the range of from 0.1 to 0.9, and 10 transformations for the range of from 0.5 to 0.9.

ECE results. Table 2 shows ECE values of all datasets and models. For CIFAR10, vector scaling, Dir-ODIR, and ReCal shows the best performance on 2/1/2 models, respectively. For CIFAR100, except LeNet5 and WRN-28-10, ReCal shows the best performance. Among our methods,

Table 2: ECE

Dataset	Model	Uncal.	TS	VS	MS-ODIR	Dir-ODIR	ReCal (z, .1-.9, 20)	ReCal (z, .5-.9, 10)	ReCal (b, .1-.9, 20)
CIFAR10	DenseNet40	0.052026	0.007037	0.004438	0.005161	**0.003943**	0.010143	0.008721	0.005892
CIFAR10	LeNet5	0.018170	0.011963	**0.009174**	0.014147	0.010525	0.011785	0.010507	0.010669
CIFAR10	ResNet110	0.045646	0.008770	0.009442	0.008829	0.008366	0.008986	**0.008206**	0.009177
CIFAR10	ResNet110 SD	0.053770	0.011407	**0.008552**	0.010187	0.009369	0.011973	0.012103	0.012845
CIFAR10	WRN 28-10	0.025076	0.009709	0.009564	0.009175	0.009429	**0.009092**	0.012459	0.010261
CIFAR100	DenseNet40	0.172838	0.015435	0.026634	0.029628	0.018949	0.015398	**0.011713**	0.018059
CIFAR100	LeNet5	**0.009991**	0.021064	0.015524	0.013149	0.014172	0.019196	0.018426	0.019367
CIFAR100	ResNet110	0.142223	0.009101	0.029982	0.034519	0.023109	0.012142	**0.008487**	0.010614
CIFAR100	ResNet110 SD	0.122932	0.009310	0.035832	0.035478	0.020747	0.009987	0.014375	**0.007918**
CIFAR100	WRN 28-10	0.053396	0.043703	0.045178	0.035509	**0.034604**	0.037270	0.035279	0.035435
ImageNet	DenseNet161	0.056384	0.019873	0.023286	0.036785	0.047707	**0.013348**	0.014474	0.016981
ImageNet	ResNet152	0.049142	0.020069	0.020672	0.034736	0.039748	0.013869	**0.013491**	0.017483
Avg.Rank		7.42	4.33	4.58	4.75	3.67	3.83	3.17	4.25

Table 3: Brier Score

Dataset	Model	Uncal.	TS	VS	MS-ODIR	Dir-ODIR	ReCal (z, .1-.9, 20)	ReCal (z, .5-.9, 10)	ReCal (b, .1-.9, 20)
CIFAR10	DenseNet40	0.013585	0.012330	0.012300	0.012256	0.012296	**0.012225**	0.012231	0.012324
CIFAR10	LeNet5	0.037836	0.037792	0.037748	0.037745	0.037706	**0.037395**	0.037403	0.037784
CIFAR10	ResNet110	0.011537	0.010439	0.010378	0.010382	0.010350	0.010322	**0.010317**	0.010441
CIFAR10	ResNet110 SD	0.015472	0.014395	0.014325	0.014231	0.014302	0.014212	**0.014140**	0.014425
CIFAR10	WRN 28-10	0.006731	0.006357	0.006380	0.006342	0.006336	**0.006300**	0.006344	0.006363
CIFAR100	DenseNet40	0.004862	0.004329	0.004346	0.004333	0.004318	0.004304	**0.004302**	0.004332
CIFAR100	LeNet5	0.007581	0.007588	0.007587	0.007580	0.007567	0.007557	**0.007543**	0.007581
CIFAR100	ResNet110	0.004521	0.004144	0.004180	0.004178	0.004149	0.004130	**0.004119**	0.004149
CIFAR100	ResNet110 SD	0.004344	0.004064	0.004046	0.004045	0.004047	0.004035	**0.004028**	0.004067
CIFAR100	WRN 28-10	0.002929	0.002915	0.002948	0.002901	**0.002898**	0.002913	0.002913	0.002926
ImageNet	DenseNet161	0.000323	0.000319	0.000316	**0.000313**	0.000324	0.000318	0.000319	0.000319
ImageNet	ResNet152	0.000305	0.000302	0.000301	**0.000299**	0.000307	0.000302	0.000302	0.000302
Avg.Rank		7.54	5.54	5.25	3.42	4.04	2.17	2.25	5.79

for most cases, (z, .5-.9, 10) shows the best performance. For ImageNet, ReCal has the best ECE for both of models; specifically, (z, .1-.9, 20) and (z, .5, .9, 20) are the best for each model.

Brier score results. Brier scores are displayed in Table 3. For CIFAR10/100, ReCal almost always shows the best performance. The only exception is when Dir-ODIR is applied to WRN 28-10 on CIFAR100. For ImageNet, MS-ODIR shows the best performance and vector scaling shows the second-best value. ReCal is slightly higher than those values. The reason that ReCal shows worse Brier score compared to vector scaling and MS-ODIR is that those two calibration methods increase the accuracy. Brier score considers both of accuracy and calibration, and the increase of accuracy results in the better Brier score.

Overall Comparison. Based on ECE results (Table 2), ReCal outperforms other algorithms on many models and dataset, and especially on ImageNet it always outperforms all other algorithms. Based on Brier score results, ReCal almost always outperforms other algorithms except a few case. With the consideration of the learning time as well, ReCal is scalable and also effective for large-scale dataset, and works well for the other medium-size dataset (CIFAR10/100) as well.

Statistical Analysis. For Table 2 and 3, we perform Friedman test. The last row of each table shows the average rank of each calibration algorithm. From the Friedman test, the p-values for each table are 0.0016 and 0.0000. Based on these p-values, we can say that the differences among the calibration algorithms are significant.

Comparison between ReCal settings. We show the three different settings of ReCal; two different transformations, and two different transformation parameter ranges for zoom-out transformation. We compare these three settings in terms of two aspects. First, between brightness and zoom-out transformation, zoom-out calibrates better, especially, on ImageNet. We conjecture that the reason is related to the fact that zoom-out is more effective in Lossy

Label-Invariant Grouping as described in Section 4. Next, for zoom-out transformations, the appropriate parameter range is related to the original image resolution. Specifically, a scale factor range of between 0.5 and 0.9 generally shows better performance on CIFAR10/100, and a scale factor range of 0.1 and 0.9 is better on ImageNet based on ECE and Brier score. Therefore, we suggest to use the small zoom-out scale factors for only large images.

Choice of Transformations. We think that a transformation type should be chosen based on the data type. For example, we think that image transformations such as zoom-out, brightness, blur, and random pixel change can be used for images, and random data drop is one example of suggested lossy label-invariant transformations for time-series data. We also conjecture that transformation parameter is connected to the data size. As shown in experimental results, for small images, it would be better to use less lossy zoom-out transformation. Similarly, it would be better to drop less frame/sample for short time-series data.

6.2.2 Learning Time

We also compute the learning time of each calibration algorithms on various datasets and models, and the result is shown in supplementary materials.

Temperature scaling is generally the fastest algorithm, and the next order is vector scaling, our method, Dir-ODIR, and MS-ODIR. Because MS-ODIR and Dir-ODIR train multiple calibration models to search the optimal hyperparameters, its calibration time is high compared to other methods. Those two algorithms train less number of calibration models for CIFAR100 compared to CIFAR10. Similarly, we reduce the number of calibration models further for ImageNet, since it is larger than the two datasets.

For ImageNet, our method takes about 51,000 seconds, or 14.1 hours for DenseNet161 and 71,000 seconds, or 19.8 hours for ResNet152. Even though this is slower than other methods like temperature scaling, and vector scaling, we think that our method can be applied to ImageNet in terms of learning time. The slowest time is 380,000 seconds, or 4.4 days for DenseNet161, and 220,000 seconds, or 2.5 days for ResNet152.

7 Conclusion

In this paper, we propose an accuracy preserving post-hoc calibration method based on a label-invariant image transformation. ReCal exploits the properties of label-invariant transformations to group inputs, and applies different temperature scaling to each group. Because ReCal is based on temperature scaling, it preserves the original classifier accuracy. In addition, it has more expressiveness compared to original temperature scaling because it uses multiple temperature scaling coefficients. Experiments on CIFAR10/100 and ImageNet datasets show that ReCal can be applied to the large-scale ImageNet, and outperforms other methods on those datasets including ImageNet.

For the future work, incorporating multiple types of transformation type may improve the calibration performance. In this paper, we use one type of transformation at a time, but, different transformation utilizes different information in example space, and combination of multiple transformation type may results in improvements. For example, we can apply brightness transformation and zoom-out transformation together, or we can also apply Gaussian blur after the transformations.

Additionally, ReCal can be extended to other types of dataset as long as appropriate transformation exists. For example, we can apply ReCal to time-series data classification. We can consider a transformation of eliminating some data at random time point. This transformation is a label-invariant transformation which decrease confidence, and ReCal can be applied to calibrate confidence. Lastly, because more accuracy-preserving post-hoc approaches have been suggesting, more comparison with such new state-of-the-art calibration algorithms will be another future work.

Acknowledgement

This work was supported in part by AFRL and DARPA FA8750-18-C-0090, ARO W911NF-20-1-0080, ONR N00014-17-1-2012, NSF-1915398 and SRC Task 2894.001. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Air Force Research Laboratory (AFRL), the Army Research Office (ARO), the Defense Advanced Research Projects Agency (DARPA), the Office of Naval Research (ONR) or the Department of Defense, or the United States Government.

References

Bahat, Y. and Shakhnarovich, G. (2020). Classification confidence estimation with test-time data-augmentation. *arXiv preprint arXiv:2006.16705*.

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3.

Chang, O., Yao, Y., Williams-King, D., and Lipson, H. (2019). Ensemble model patching: A parameter-efficient variational bayesian neural network. *arXiv preprint arXiv:1905.09453*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059.

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. *arXiv preprint arXiv:1706.04599*.

Gupta, K., Rahimi, A., Ajanthan, T., Mensink, T., Sminchisescu, C., and Hartley, R. (2020). Calibration of neural networks using splines. *arXiv preprint arXiv:2006.12800*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

Huang, G., Sun, Y., Liu, Z., Sedra, D., and Weinberger, K. Q. (2016). Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer.

Khan, M. E., Nielsen, D., Tangkaratt, V., Lin, W., Gal, Y., and Srivastava, A. (2018). Fast and scalable bayesian deep learning by weight-perturbation in adam. *arXiv preprint arXiv:1806.04854*.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

Kull, M., Nieto, M. P., Kängsepp, M., Silva Filho, T., Song, H., and Flach, P. (2019). Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In *Advances in Neural Information Processing Systems*, pages 12295–12305.

Kumar, A., Sarawagi, S., and Jain, U. (2018). Trainable calibration measures for neural networks from kernel mean embeddings. In *International Conference on Machine Learning*, pages 2805–2814.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pages 6402–6413.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lee, K., Lee, H., Lee, K., and Shin, J. (2017). Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Müller, R., Kornblith, S., and Hinton, G. E. (2019). When does label smoothing help? In *Advances in Neural Information Processing Systems*, pages 4696–4705.

Naeini, M. P., Cooper, G. F., and Hauskrecht, M. (2015). Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, volume 2015, page 2901. NIH Public Access.

Patel, K., Beluch, W., Yang, B., Pfeiffer, M., and Zhang, D. (2020). Multi-class uncertainty calibration via mutual information maximization-based binning. *arXiv preprint arXiv:2006.13092*.

Patel, K., Beluch, W., Zhang, D., Pfeiffer, M., and Yang, B. (2019). On-manifold adversarial data augmentation improves uncertainty calibration. *arXiv preprint arXiv:1912.07458*.

Platt, J. et al. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

Rahimi, A., Shaban, A., Cheng, C.-A., Boots, B., and Hartley, R. (2020). Intra order-preserving functions for calibration of multi-class neural networks. *arXiv preprint arXiv:2003.06820*.

Rajpurkar, P., Irvin, J., Ball, R. L., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C. P., et al. (2018). Deep learning for chest radiograph diagnosis: A retrospective comparison of the chexnext algorithm to practicing radiologists. *PLoS medicine*, 15(11):e1002686.

Seo, S., Seo, P. H., and Han, B. (2019). Learning for single-shot confidence calibration in deep neural networks through stochastic inferences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9030–9038.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *nature*, 550(7676):354–359.

Thulasidasan, S., Chennupati, G., Bilmes, J. A., Bhattacharya, T., and Michalak, S. (2019). On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *Advances in Neural Information Processing Systems*, pages 13888–13899.

Tran, G.-L., Bonilla, E. V., Cunningham, J., Michiardi, P., and Filippone, M. (2019). Calibrating deep convolutional gaussian processes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1554–1563.

Veit, A., Wang, Y., and Chen, D. (2017). A pytorch implementation for densely connected convolutional networks (densenets).

Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., and Bengio, Y. (2019). Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR.

Wang, Y., Chen, T., Xu, H., Ding, S., Lv, H., Shao, Y., Peng, N., Xie, L., Watanabe, S., and Khudanpur, S. (2019). Espresso: A fast end-to-end neural speech recognition toolkit. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 136–143. IEEE.

Wenger, J., Kjellström, H., and Triebel, R. (2020). Non-parametric calibration for classification. In *International Conference on Artificial Intelligence and Statistics*, pages 178–190. PMLR.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500.

Yang, W., Zhang, H., and Li, Z. a. (2019). Classification with pytorch.

Zadrozny, B. and Elkan, C. (2001). Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml*, volume 1, pages 609–616. Citeseer.

Zadrozny, B. and Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699.

Zagoruyko, S. and Komodakis, N. (2016). Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Zhang, G., Sun, S., Duvenaud, D., and Grosse, R. (2017). Noisy natural gradient as variational inference. *arXiv preprint arXiv:1712.02390*.

Zhang, J., Kailkhura, B., and Han, T. (2020). Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. *arXiv preprint arXiv:2003.07329*.