Abstract

Deep Neural Networks (DNNs), despite their tremendous success in recent years, could still cast doubts on their predictions due to the intrinsic uncertainty associated with their learning process. Ensemble techniques and post-hoc calibrations are two types of approaches that have individually shown promise in improving the uncertainty calibration of DNNs. However, the synergistic effect of the two types of methods has not been well explored. In this paper, we propose a truth discovery framework to integrate ensemble-based and post-hoc calibration methods. Using the geometric variance of the ensemble candidates as a good indicator for sample uncertainty, we design an accuracy-preserving truth estimator with provably no accuracy drop. Furthermore, we show that post-hoc calibration can also be enhanced by truth discovery-regularized optimization. On large-scale datasets including CIFAR and ImageNet, our method shows consistent improvement against state-of-the-art calibration approaches on both histogram-based and kernel density-based evaluation metrics. Our code is available at https://github.com/horsepurve/truly-uncertain.

1 INTRODUCTION

We live in an uncertain world. With the increasing use of deep learning in the real world, quantitative estimation of the predictions from deep neural networks (DNNs) must not be neglected, especially when it comes to medical imaging [Esteva et al., 2017] [Ma et al., 2019], disease diagnosis [De Fauw et al., 2018] [Ma et al., 2018], and autonomous driving [Kendall et al., 2017]. Uncertainty also plays an important role in differentially private data analysis [Bassily et al., 2013].

Modern deep neural networks, despite their extraordinary performance, are oft-criticized as being poorly calibrated and prone to be overconfident, thus leading to unsatisfied uncertainty estimation. The process of adapting deep learning’s output to be consistent with the actual probability is called uncertainty calibration [Guo et al., 2017], and has drawn a growing attention in recent years.

For a better calibration of the uncertainty of DNNs, the efforts to date have been concentrated on developing more effective calibration and evaluation methods. Existing calibration methods roughly fall into two categories, depending on whether an additional hold-out calibration dataset is used. (1) Post-hoc calibration methods use a calibration dataset to learn a parameterized transformation that maps from classifiers’ raw outputs to their expected probabilities. Quite a few techniques in this category can be used to learn the mapping, such as Temperature Scaling (TS) [Guo et al., 2017] [Kull et al., 2019], Ensemble Temperature Scaling (ETS) [Zhang et al., 2020], and cubic spline [Gupta et al., 2021], etc. However, the expressivity of the learnable mapping could still be limited in all of them. This is evidenced by the fact that in TS a single temperature parameter $T$ is tuned, while ETS brings in three additional ensemble parameters. Thus, it is desirable to explore a more sophisticated form of the mapping function. (2) Another line of methods adapt the training process so that the predictions are better calibrated. Techniques in this category include mixup training [Thulasidasan et al., 2019], pre-training [Hendrycks et al., 2019a], label-smoothing [Müller et al., 2019], data augmentation [Ashukha et al., 2020], self-supervised learning [Hendrycks et al., 2019b], Bayesian approximation [Gal and Ghahramani, 2016] [Gal et al., 2017], and Deep Ensemble (DE) [Lakshminarayanan et al., 2017], with its variants (Snapshot Ensemble [Huang et al., 2017a], Fast Geometric Ensemble (FGE) [Garipov et al., 2018], SW-A-Gaussian (SWAG) [Maddox et al., 2019]). Methods from these two categories thrive in recent years, and a natural idea is to combine them together. Recently, Ashukha et al. [2020] points out the
Accordingly, in this paper, we propose truth discovery as an ideal tool for improving uncertainty calibration of deep learning, and make several contributions as follows:

1. We propose Truth Discovery Ensemble (TDE) that improves Deep Ensemble, and show that model uncertainty can be easily derived from truth discovery.

2. Considering that uncertainty calibration approaches may potentially cause a diminished accuracy, we further develop a provably accuracy-preserving Truth Discovery Ensemble (aTDE) via geometric optimization.

3. We propose an optimization approach that directly minimizes ECEs, works for both histogram-based and KDE-based metrics, and integrates multiple metrics via compositional training.

4. We further incorporate the discovered information (i.e. Entropy based Geometric Variance) into the post-hoc calibration pipeline (pTDE) and elevate the performance to a higher level.

To summarize, we show how truth discovery can benefit both ensemble-based and post-hoc uncertainty calibrations, and validate our proposed methods via experiments upon large-scale datasets, using both binning-based and binning-free metrics, along with comprehensive ablation studies.

2 PRELIMINARIES OF UNCERTAINTY CALIBRATION

For an arbitrary multi-class classifier (not necessarily neural network) \( f_\theta : \mathcal{D} \subseteq \mathbb{R}^d \rightarrow \mathcal{Z} \subseteq \Delta^L \) that can make \( L \) predictions for \( L \) classes, its outputs (in any scale) can be transformed into a "probability vector" \( z \in \mathcal{Z} \) such that:

\[
\sum_{l=1}^{L} z_l = 1, \quad 0 \leq z_l \leq 1.
\]  

This can be done by the softmax function, which usually tails the last layer of a deep neural network. Here, \( \Delta^L \) is the probability simplex in \( L \) dimensional space. Note that the classifier parameters can also be drawn from a distribution \( \theta \sim q(\theta) \), e.g., ResNets with random initialization being the only difference.

Although \( z \) is in the probability simplex \( \Delta^L \), its components may not necessarily have anything to do with, but sometimes are misinterpreted as, the probability of each class. Similarly, the maximum value of the \( L \) outputs, \( \max_l z_l \), was used to represent the "confidence" that the classifier has on its prediction. To avoid possible misleading, \( \max_l z_l \) is referred to as winning score \( v \) (i.e., \( v = \max_l z_l \) ) [Thulasidasan et al., 2019] hereinafter.

For both ensemble-based and post-hoc calibration methods, the model is trained based on a set of \( N \) training samples \( \{ (x(i), y(i)) \}^{N}_{i=1}, x(i) \in \mathcal{D}, y(i) \in \{1, ..., L\} \). Let random variables \( X, Y \) represent input data and label, respectively. Then, another random variable \( Z = f_\theta(X) \) stands for the
probability vector. If $z_l$ indeed represents the actual probability of class $l$ (which usually not), then, the following should hold:

$$P(Y = l | Z = z) = z_l.$$  \hspace{1cm} (2)

At this time, we also call the classifier $f_\theta$ to be perfectly calibrated. It is well known that the probabilities $P(Y = l | Z = z)$ are hard to evaluate, since there is no ground-truth for the probability of an input $x^{(i)}$ being misclassified as class $l \neq y^{(i)}$. In this paper, we focus on a variant of Eq. (2), which only measures the probability of the sample being correctly classified:

$$P(Y = y^{(i)} | Z = z^{(i)}) = v^{(i)}$$ \hspace{1cm} (3)

where $v^{(i)}$ is the winning score which is also the only value taken into consideration when evaluate top-1 accuracy (ACC).

**Ensemble of Deep Neural Networks.** Although it is computationally demanding, ensemble (Deep Ensemble [Lakshminarayanan et al., 2017], Snapshot Ensemble [Huang et al., 2017a], etc.) remains as a popular approach for uncertainty calibration. Formally, for a classifier $f_\theta$ with parameter distribution $q(\theta)$, the prediction of sample $x^{(i)}$ is given by:

$$z^{(i)}_{\text{ens}} = \int f_\theta(x^{(i)})) q(\theta) d\theta$$ \hspace{1cm} (4)

which can be approximated by $S$ independently trained classifiers as $\frac{1}{S} \sum_{s=1}^{S} f_{\theta^{(s)}}(x^{(i)}), \theta^{(s)} \sim q(\theta).$ The $S$ classifiers can be obtained either by independent random initialization (Deep Ensemble) or periodically conversion into local minimum via learning rate decay (Snapshot Ensemble). Since each $z^{(i)}_{\text{ens}} \in \Delta^L$, $z^{(i)}_{\text{ens}}$ is also on the probability simplex.

**Post-hoc Calibration of Deep Neural Networks.** Until now, we have not yet introduced the concept of confidence, since all non-post-hoc approaches take the winning score as a representation of the confidence, based on which the calibration error is subsequently measured. In post-hoc calibration, on the other hand, a set of hold-out calibration samples $(x^{(i)}, y^{(i)})_{i=1}^{N_c}$ is required to learn a mapping from $z$ to another probability vector $\pi = \mathcal{T}(z)$ with a learnable function $\mathcal{T} : \Delta^L \rightarrow \Delta^L$, and $\max_i \pi_i$ is referred to as confidence $w$ in this paper, i.e. $w = \max_i \pi_i.$ Note that $\arg \max_i \pi_i$ is not necessarily equal to $\arg \max_i z_i$, and at this time the calibration may potentially decrease the accuracy, if the latter is the correct class. With this, our goal Eq. (3) now becomes $P(Y = y^{(i)} | Z = z^{(i)}) = w^{(i)}$. Since we tackle both post-hoc and non-post-hoc calibrations in this paper, we distinguish winning score and confidence explicitly in that the former directly comes from the classifier $f_\theta$ while the latter is derived from a mapping deliberately learned for confidence modeling.

**Calibration Error Evaluation.** For the evaluation of a calibration algorithm, the calibration function $\mathcal{T}$ is applied on another evaluation dataset $(x^{(i)}, y^{(i)})_{i=1}^{N_e}$ of size $N_e$ which has no overlapping with neither the training nor the calibration datasets. In histogram-based evaluation metric, the $N_e$ samples are split into $B$ predefined bins. Formally, we define $B$ pairs of endpoints $\{(\mu_b, \nu_b)\}_{b=1}^{B}$, $\mu_b = \mu_{b+1}$, and $B$ point sets $\{P_b\}_{b=1}^{B}$ such that $\mu_b \leq w < \nu_b, \forall w \in P_b$. Then, the Expected Calibration Error (ECE) is defined as:

$$ECE(f_\theta) = \sum_{b=1}^{B} \frac{|P_b|}{K} |ACC(P_b) - con.f(P_b)|,$$ \hspace{1cm} (5)

which measures the empirical deviation of the sample accuracy in the $b^{th}$ bin: $ACC(P_b) = \frac{1}{|P_b|} \sum_{j=1}^{|P_b|} 1(\arg \max_i z_i^{(j)} = y^{(j)})$ and the average confidence in it: con.f($P_b$) = $\frac{1}{|P_b|} \sum_{j=1}^{|P_b|} w^{(j)}$. The indicator function $1 : B \rightarrow \{0, 1\}$ returns 1 if the Boolean expression is true and otherwise 0.

The ECE metric can be easily affected by the number of bins $B$ and the positions of the endpoints. Without the use of binning, $ECE_{\text{KDE}}$ estimates the calibration error by a kernel function $K : \mathbb{R} \rightarrow \mathbb{R}_0+$ with bandwidth $h > 0$. Based on Bayesian rule, $ECE_{\text{KDE}}$ is given as:

$$ECE_{\text{KDE}}(f_\theta) = \int |w - \tilde{ACC}(w)| \tilde{P}(w) dw$$ \hspace{1cm} (6)

in which $\tilde{ACC}(w)$ is the expected accuracy if the sample confidence is $w$. $\tilde{P}(w)$ and $\tilde{ACC}(w)$ are determined by kernel density estimation as:

$$\tilde{P}(w) = \frac{h^{-1}}{N_e} \sum_{i=1}^{N_e} K_h(w - w^{(i)}),$$  

$$\tilde{ACC}(w) = \frac{\sum_{i=1}^{N_e} 1(\arg \max_i z_i^{(i)} = y^{(i)}) K_h(w - w^{(i)})}{\sum_{i=1}^{N_e} K_h(w - w^{(i)})}.$$  

3 TRUTH DISCOVERY ENSEMBLE

Existing ensemble techniques in deep learning take the average of predictions made by multiple classifiers derived either from random initialization (Deep Ensemble [Lakshminarayanan et al., 2017]), periodically learning rate decay (Snapshot Ensemble [Huang et al., 2017a]), or from connected optima on the loss functions (Fast Geometric Ensembling [Garipov et al., 2018]), but scarcely utilize the sample level variance among the members of an ensemble. To make use of such information, we first introduce a vanilla truth discovery algorithm in Deep Ensemble context, and then extend it to one with accuracy-preserving guarantee.
3.1 TRUTH DISCOVERY WITHIN PROBABILITY SIMPLEX

To be consistent with truth discovery literature, we use sources to denote the $S$ independently trained models. For every sample $(x^{(i)}, y^{(i)})$ in the evaluation dataset, $S$ independent predictions $z^{(s), (i)} = f_{g(s)}(x^{(i)})$ are made from all $S$ sources (denoted by $z_s$ hereinafter for brevity). Since the classifiers were trained with stochastic gradient descent (SGD), they may make wrong decisions on every $x^{(i)}$. To model such a behavior, we assign a reliability value $\omega_s$ to each classifier $f_{g(s)}$.

Definition 3.1 ([Li et al. 2014]). Given the set of points $\{z_s\}_{s=1}^S \subseteq \Delta^L$ from $S$ classifiers, truth discovery aims at finding the truth probability vector $z^* \in \Delta^L$ and meanwhile the reliability $\omega_s$ for the $s^{th}$ classifier, such that the following objective function is minimized:

$$\min_{z^*: \{\omega_s\}_{s=1}^S} \sum_{s=1}^S \omega_s ||z^* - z_s||^2$$

s.t. $\sum_{s=1}^S e^{-\omega_s} = 1.$

(7)

With this definition, interestingly, we can show a direct relationship between truth discovery and model uncertainty. We additionally define uncertainty of source, $\upsilon_s$, as the opposite of source reliability, i.e. $\upsilon_s = e^{-\omega_s}$. Then, (7) can be written as:

$$\min_{z^*: \{\upsilon_s\}_{s=1}^S} \sum_{s=1}^S -||z^* - z_s||^2 \ln \upsilon_s$$

s.t. $\sum_{s=1}^S \upsilon_s = 1.$

(8)

This is essentially the Cross Entropy (CE) of source uncertainty $\upsilon_s$ and $||z^* - z_s||^2/2$, which is the similarity between the optimum probability vector to each source vector ($0 \leq \upsilon_s \leq 1, 0 \leq ||z^* - z_s||^2/2 \leq 1$). Thus, the minimization process is to ensure that the solution resolves the ambiguity of the system as much as possible. Hence, truth discovery can ideally benefit uncertainty calibration through finding the truth vector.

Algorithms for approximating the global optimum exist [Ding and Xu 2020; Huang et al. 2019]. But here with the assumption Eq. (1) that all the possible truth vectors fall on the probability simplex $\Delta^L$, we adopt a simpler solution. Since both the truth vector $z^*$ and source reliabilities are unknown, we can alternatively update the reliability/uncertainty and the truth vector. Specifically, if $z^*$ is temporarily fixed, the optimum reliability values can be found through Lemma 3.1.

Lemma 3.1 ([Li et al. 2014]). If $z^*$ is fixed, the following reliability value for each source $\omega_s$ minimizes the objective

$$\min_{\omega_s} \sum_{s=1}^S \omega_s ||z^* - z_s||^2$$

$$\text{s.t. } \sum_{s=1}^S e^{-\omega_s} = 1.$$
however, this requirement may not be satisfied, since \( z^* \) could change the predictions. Suppose that \( c \) is the predicted class derived from ensemble \( z_{ens} \), i.e., \( c = \arg \max_l(z_{ens}) \). After the truth vector \( z^* \) is found, we want to at least maintain the accuracy of ensemble. Thus, Eq. (7) can be retrofitted to be an accuracy-preserving version.

**Definition 3.2** (Accuracy-preserving truth discovery). Given the set of points \( \{z_s\}_{s=1}^S \subseteq \Delta_L \) and the ensemble vector \( z_{ens} \), find the truth vector \( z^* \in \Delta_L \) and reliabilities \( \{\omega_s\}_{s=1}^S \) such that the following objective function is minimized:

\[
\begin{align*}
\text{minimize} & \quad z^*, \omega \quad \sum_{s=1}^S \omega_s \|z^* - z_s\|^2 \\
\text{s.t.} & \quad \sum_{s=1}^S e^{-\omega_s} = 1 \\
& \quad \arg \max_l z^*_l = \arg \max_l (z_{ens})_l.
\end{align*}
\]

It can be formulated as a geometric optimization problem. See Figure 1 for an illustration when \( L = 3 \). The constraint \( \arg \max_l z^*_l = \arg \max_l (z_{ens})_l \) introduces a subspace \( \Omega : z_l > z_m, \forall m \neq l \). The discovery of the truth vector must be performed within the accuracy-preserving simplex \( \Delta_\alpha = \Delta_L \cap \Omega \). Intuitively, when \( z^* \) falls outside of the accuracy-preserving simplex \( \Delta_\alpha \) in one iteration, we can find the projection of \( z^* \) onto \( \Delta_\alpha \) to pull it back to \( \Delta_\alpha \). This projection can be found by Algorithm 2 which is proved in Theorem 3.1. When the projection is done, Algorithm 1 continues until the desired truth vector is found.

**Theorem 3.1.** Algorithm 2 preserves the accuracy of the prediction.

**Proof.** The theorem can be proved using Lagrange Multipliers with Karush-Kuhn-Tucker (KKT) Conditions, see supplementary material Sec. A.1 for details.

---

**Algorithm 2:** Projection from truth vector \( z \) onto accuracy-preserving simplex \( \Delta_\alpha \).

**Data:** \( z \)

**Result:** \( \tilde{z} \)

1. \( c \leftarrow \arg \max_l (z_{ens})_l; \quad \tilde{z} \leftarrow z; \)
2. \( M \leftarrow \text{ARGSORT} \{ (z_1, ..., z_L) \}; \)
3. if \( M[1] = c \) then
4. return \( z \)
5. else
6. for \( l \leftarrow 1, ..., L \) do
7. \( \tilde{z} \leftarrow \frac{1}{M[1]} (z_c + z_{M[1]} + ... + z_{M[l]}); \)
8. if \( \tilde{z} > z_{M[l+1]} \) then
9. \( z_c \leftarrow \tilde{z}; \quad z_{M[n]} \leftarrow \tilde{z}, \forall n \leq l; \)
10. return \( \tilde{z} \)

---

### 4 TRUTH DISCOVERY-REGULARIZED POST-HOC CALIBRATION

Although ECE-like scores are difficult to be optimized directly, recent works have attempted to minimize ECE either by using maximum mean calibration error (MMCE), a kernelized version of the ECE [Kumar et al., 2018], or by rank preserving transforms [Bai et al., 2021]. To provide a better solution, we formulate the minimization of ECE (and also ECE\textsuperscript{KDE}) as an optimization problem in high dimensions, and show how it can be easily extended to incorporate the information gained from truth discovery.

**Optimization of ECEs.** For simplicity, we only consider the confidence for the ground-truth (namely, top-1) class, which is essentially the probability of a sample being correctly predicted. Then, our learnable mapping becomes \( w = \mathcal{T}(z) : \Delta_L \rightarrow \mathbb{R} \). One step further, if only the winning score is considered, then \( w = \mathcal{T}(v) : \mathbb{R} \rightarrow \mathbb{R} \). Next, we find the specific form of \( \mathcal{T} \). It has been shown that deep neural networks tends to be overconfident on most of the predictions [Guo et al., 2017]. Inspired by this, we impose an attenuation factor \( \varphi(v) \) on every sample, which is a function of the winning score \( v \) so that the adjusted confidence becomes \( w = v - \varphi(v) \). The simplest form of \( \varphi(v) \) is to use a constant within a bin \( P_k \). Thus, we define an attenuation weight \( \psi_b \) for the bin \( P_k \). All the attenuation weights \( \{\psi_b\}_{b=1}^B \) can be viewed as a point \( \psi \in \mathbb{R}^B \). Now we have the definition of the mapping function:

\[
\begin{align*}
&w = v - \varphi(v) = v - \psi_k, \\
&\text{s.t.} \quad \mu \leq v < \nu_k.
\end{align*}
\]

Notice that for consistency we call \( \varphi(v) \) the attenuation factor, but it can also enhance the confidence \( w \) if \( \varphi(v) < 0 \) somewhere.

Now our goal is to find the location of \( \psi \) in \( \mathbb{R}^B \) such that the expected calibration error is minimized:

\[
\begin{align*}
\text{minimize} & \quad ECE(\{P_b\}_{b=1}^B), \\
\text{s.t.} & \quad \psi_b^B \leq v < \psi_b^{B+1}.
\end{align*}
\]

where \( ECE \) is shown in Eq. (5). Since all the computations in ECE (and ECE\textsuperscript{KDE}) are differentiable, the minimization of ECEs can be done by gradient descent methods. Here, a mini-batch Stochastic Gradient Descent (SGD) approach is used. In each epoch, a subset of calibration data is sampled, based on which the attenuation weights are updated. The optimization process is encapsulated in Algorithm 3, in which ECE and ECE\textsuperscript{KDE} can be used interchangeably. The algorithm can be easily implemented by virtue of automatic differentiation libraries e.g. PyTorch [Paszke et al., 2019].

**Compositional Approach for the Optimization of ECEs.** Even an accelerated method for KDE computation is used [O’Brien et al., 2016]. KDE-based metric is still much more
Table 1: Comparison of TDE/aTDE with Deep Ensemble (DE) before post-hoc calibration.

| Model       | Method | ECE$^{KDE}$↓ | ECE$^{KDE}$↑ | ECE↓ | ECE↑ | ACC↑ | ECE$^{KDE}$↓ | ECE$^{KDE}$↑ | ECE↓ | ECE↑ | ACC↑ |
|-------------|--------|--------------|--------------|------|------|------|--------------|--------------|------|------|------|
| CIFAR10     |        |              |              |      |      |      |              |              |      |      |      |
| PreResNet110| DE     | 2.88         | 2.50         | 82.83| 3.06 | 2.58 | 83.02        | 1.31         | 0.48 | 96.38| 1.15 | 0.43 |
|             | TDE    | 1.60         | 1.98         | 82.89| 1.81 | 1.94 | 83.01        | 1.05         | 0.75 | 96.38| 1.01 | 0.59 |
|             | aTDE   | 1.55         | 1.92         | 82.83| 1.78 | 1.95 | 83.02        | 1.07         | 0.75 | 96.38| 1.02 | 0.60 |
| PreResNet164| DE     | 2.39         | 2.09         | 83.52| 2.46 | 2.00 | 83.55        | 1.32         | 0.34 | 96.68| 1.31 | 0.35 |
|             | TDE    | 1.31         | 1.72         | 83.49| 1.44 | 1.61 | 83.54        | 1.14         | 0.65 | 96.66| 1.08 | 0.52 |
|             | aTDE   | 1.33         | 1.76         | 83.52| 1.42 | 1.62 | 83.55        | 1.13         | 0.63 | 96.68| 1.08 | 0.53 |
| WideResNet  | DE     | 6.45         | 5.85         | 84.38| 6.40 | 5.78 | 84.28        | 1.06         | 0.35 | 97.17| 1.20 | 0.41 |
|             | TDE    | 5.48         | 5.03         | 84.37| 5.58 | 5.07 | 84.29        | 1.14         | 0.49 | 97.15| 1.10 | 0.40 |
|             | aTDE   | 5.49         | 5.05         | 84.38| 5.57 | 5.06 | 84.28        | 1.13         | 0.47 | 97.17| 1.12 | 0.41 |
| ImageNet    | DE/SE  | 3.11         | 3.12         | 79.25| 3.24 | 3.17 | 79.37        | 1.85         | 2.11 | 78.46| 1.73 | 2.03 |
|             | TDE    | 2.16         | 2.42         | 79.22| 2.41 | 2.58 | 79.35        | 1.69         | 2.08 | 78.45| 1.66 | 2.10 |
|             | aTDE   | 2.19         | 2.45         | 79.25| 2.43 | 2.60 | 79.37        | 1.70         | 2.10 | 78.46| 1.69 | 2.13 |

Algorithm 3: Optimization of ECEs.

**Data:** \{(v(i), y(i))\}_{i=1}^{N_c}

**Result:** \{\psi_b\}_{b=1}^{B}

1. for epoch ← 1, ..., #epoch do
2. sample \{(v(j), y(j))\}_{j=1}^{N_c} \sim \{(v(i), y(i))\}_{i=1}^{N_c};
3. w = v − φ(v) ; ⑤ apply attenuation factor
4. loss ← ECE or ECE$^{KDE}$ ; ⑥ forward propagation
5. update \{\psi_b\}_{b=1}^{B} ; ⑦ backward propagation
6. return \{\psi_b\}_{b=1}^{B}

Algorithm 3 is time-consuming compared to histogram-based metrics. A natural question that arises here is whether the minimization of ECE$^{KDE}$ can be speeded up by the minimization of ECE. To answer this, we first find the attenuation weights by using ECE as the loss function, and then use the obtained \{\psi_b\}_{b=1}^{B} as an initial guess for the minimization of ECE$^{KDE}$. This approach enables the compositional optimization of histogram-based and KDE-based calibration errors.

**ECE Optimization Regularized by Truth Discovery.**

Given a discovered truth vector \(z^*\), let \(V\) denote the total squared distance to \(z^*\) (i.e., \(V = \sum_s ||z^* - z_s||^2\)) and \(q_s\) denote the contribution of each \(z_s\) to \(V\) (i.e., \(q_s = \frac{||z^* - z_s||^2}{V}\)). Then, the entropy induced by \{\lambda_{s}\}_{s=1}^{S} is:

\[
H = -\sum_{s=1}^{S} q_s \log q_s = \frac{1}{V} \sum_{s=1}^{S} ||z^* - z_s||^2 \log \frac{V}{||z^* - z_s||^2}.
\]

(14)

Based on these, we can define the Entropy based Geometric Variance (HV):

**Definition 4.1** (Entropy based Geometric Variance [Ding and Xu 2020]). Given the point set \{\mathbf{z}_s\}_{s=1}^{S} and a point \(z^*\), the Entropy based Geometric Variance (HV) is \(H \times V\) where \(H \) and \(V \) are defined as shown above.

With this definition, it is easy to see that the objective function of truth discovery \(L\) is exactly the entropy based geometric variance (HV) and the optimization problem \(\min_{\mathbf{z}} L = \min_{\mathbf{z}} H \times V\) is equivalent to finding a point \(z^*\) to minimize \(HV\).

If the truth vector \(z^*\) has been determined, then \(HV\) is an indicator of the ambiguity of the system, and can be borrowed as an external information for our ECE optimization. Despite being in the same bin and overconfident, the sample confidences should not be attenuated at the same scale. Instead, the sample with higher \(HV\) (i.e. higher variance and uncertainty) is to be attenuated by a larger magnitude. Consequently, the mapping function can be reshaped as:

\[
w(i) = v(i) - \varphi(v(i)) = v(i) - \alpha_1 \psi_c - \alpha_2 HV(i), \quad (15)
\]

where \(\alpha_1, \alpha_2\) are hyperparameters, and \(HV(i)\) is essentially the value of the objective function of truth discovery as computed in Section 3.2 for each sample \((x(i), y(i))\) in a total of \(N_c + N_e\) calibration and evaluation samples. The learning of the mapping function \(\mathcal{T}\) from the calibration data is a supervised learning problem. Hence, \(\mathcal{T}\) is expected to be overfitted to calibration data. By incorporating orthogonal information (i.e. \(HV\)) acquired from the truth discovery of multiple ensemble classifiers, we can learn a mapping \(\mathcal{T}\) that generalizes better on the evaluation datasets.

**5 EXPERIMENTS**

The main goals of our experiments are to: (1) compare Truth Discovery Ensemble (TDE), especially the accuracy-preserving version (aTDE), with ensemble-based calibration
Table 2: Comparison of post-hoc calibration methods. The best results are highlighted in bold. We also underline the best results excluding our method. ECEs are reported in terms of mean±standard deviation obtained from 5 random replications.

| Dataset     | Model      | ECE % | ECE KDE % |
|-------------|------------|-------|-----------|
|             | DE         | TS    | ETS       | IRM | pTDE | DE       | TS    | ETS    | IRM | pTDE |
| CIFAR100    | ResNet18   | 2.86±0.42 | 2.70±0.36 | 2.31±0.30 | 2.52±0.35 | 1.64±0.37 | 2.41±0.35 | 2.37±0.37 | 2.25±0.33 | 2.09±0.36 | 1.63±0.37 |
| CIFAR100    | DenseNet121| 1.69±0.15 | 1.68±0.16 | 1.45±0.15 | 1.70±0.19 | 1.29±0.19 | 1.67±0.07 | 1.66±0.08 | 1.65±0.12 | 1.66±0.13 | 1.49±0.12 |
| CIFAR100    | ResNeXt29  | 1.92±0.26 | 1.93±0.27 | 1.54±0.22 | 1.88±0.36 | 1.20±0.34 | 1.89±0.12 | 1.86±0.07 | 1.87±0.11 | 1.74±0.24 | 1.63±0.12 |
| CIFAR10     | ResNet18   | 0.94±0.14 | 0.65±0.13 | 0.65±0.13 | 0.67±0.24 | 0.75±0.10 | 1.50±0.11 | 1.48±0.06 | 1.48±0.06 | 1.35±0.08 | 1.21±0.18 |
| CIFAR10     | DenseNet121| 0.88±0.11 | 0.65±0.13 | 0.65±0.13 | 0.55±0.06 | 0.50±0.14 | 1.78±0.17 | 1.76±0.16 | 1.76±0.16 | 1.56±0.15 | 1.45±0.20 |
| CIFAR10     | ResNeXt29  | 0.46±0.09 | **0.33±0.10** | 0.34±0.10 | 0.45±0.13 | 0.39±0.15 | 1.38±0.06 | 1.43±0.09 | 1.43±0.09 | 1.42±0.10 | **1.36±0.06** |
| ImageNet    | 10 (DE)    | 3.06±0.12 | 1.59±0.09 | **0.96±0.07** | 1.93±0.16 | **0.88±0.11** | 3.13±0.11 | 1.16±0.14 | 1.06±0.11 | 1.79±0.11 | **0.93±0.08** |
| ImageNet    | 30 (DE)    | 3.19±0.06 | 1.47±0.15 | **0.87±0.10** | 1.89±0.12 | **0.81±0.16** | 3.26±0.05 | 1.10±0.12 | 0.97±0.09 | 1.75±0.07 | **0.95±0.11** |
| ImageNet    | 10 (SE)    | 2.32±0.09 | 1.56±0.05 | **1.04±0.10** | 1.74±0.04 | **0.71±0.17** | 2.18±0.09 | 1.14±0.06 | 1.07±0.08 | 1.51±0.07 | **0.95±0.07** |
| ImageNet    | 30 (SE)    | 2.13±0.07 | 1.57±0.07 | **0.95±0.15** | 1.70±0.07 | **0.76±0.22** | 1.89±0.09 | 1.16±0.08 | 0.95±0.17 | 1.43±0.11 | **0.87±0.22** |

Table 3: Ablation Study of the proposed truth-discovery-regularized post-hoc calibration (pTDE). For the four variants of pTDE, the blue/red color denotes if compositional training (Comp.)/truth-discovery regularization (Truth. Reg.) is utilized.

| Dataset     | Model      | ECE | ECE KDE |
|-------------|------------|-----|---------|
|             | opt KDE    | pTDE | pTDE KDE |
|             | opt | opt KDE | pTDE | pTDE KDE | opt | opt KDE | pTDE | pTDE KDE |
| **Comp.**   | **Truth. Reg.** | X | X | X | X | X | X | X | X | X |
| CIFAR100    | ResNet18   | 1.17±0.22 | 1.68±0.38 | **1.59±0.37** | 1.64±0.37 | 1.76±0.33 | 1.73±0.31 | 1.71±0.35 | **1.63±0.37** |
| CIFAR100    | DenseNet121| **1.29±0.17** | 1.31±0.15 | 1.29±0.20 | 1.29±0.19 | 1.56±0.09 | 1.53±0.11 | 1.52±0.09 | **1.49±0.12** |
| CIFAR100    | ResNeXt29  | 1.27±0.33 | 1.24±0.35 | **1.19±0.35** | 1.20±0.34 | 1.68±0.12 | 1.67±0.12 | 1.64±0.11 | **1.63±0.12** |
| CIFAR10     | ResNet18   | 0.65±0.20 | 0.72±0.15 | **0.65±0.17** | 0.75±0.10 | 1.33±0.12 | 1.26±0.18 | 1.31±0.12 | **1.21±0.18** |
| CIFAR10     | DenseNet121| 0.45±0.06 | 0.46±0.16 | **0.44±0.04** | 0.50±0.14 | 1.54±0.16 | 1.46±0.17 | 1.55±0.18 | **1.45±0.20** |
| CIFAR10     | ResNeXt29  | 0.41±0.10 | 0.41±0.13 | 0.40±0.11 | **0.39±0.15** | 1.37±0.07 | 1.36±0.07 | 1.37±0.06 | **1.36±0.06** |
| ImageNet    | 10 (DE)    | **0.76±0.13** | 0.83±0.13 | 0.81±0.09 | 0.88±0.11 | 0.99±0.08 | **0.92±0.07** | 1.01±0.08 | 0.93±0.08 |
| ImageNet    | 30 (DE)    | 0.88±0.13 | 0.85±0.11 | 0.87±0.16 | **0.81±0.16** | 1.09±0.12 | 1.02±0.09 | 1.03±0.10 | **0.95±0.11** |
| ImageNet    | 10 (SE)    | 0.80±0.19 | 0.73±0.19 | 0.75±0.19 | **0.71±0.17** | 1.04±0.12 | 0.96±0.09 | 1.01±0.11 | **0.95±0.07** |
| ImageNet    | 30 (SE)    | 0.81±0.16 | **0.76±0.20** | 0.78±0.17 | 0.76±0.22 | 0.98±0.19 | 0.90±0.22 | 0.95±0.22 | **0.87±0.22** |
methods; (2) for post-hoc calibration scheme, compare truth discovery-regularized calibration (pTDE) with state-of-the-art methods on a wide range of network architectures and datasets; investigate if different components of our methods collaboratively contribute to the overall elevation of performance by ablation studies.

5.1 IMPROVED DEEP ENSEMBLE BY TRUTH DISCOVERY

Experimental Setup. For a fair comparison, we downloaded the trained models of Ashukha et al. [2020] including PreResNet110/164 [He et al. 2016] and WideResNet28x10 [Zagoruyko and Komodakis 2016] trained on CIFAR10/100 [Krizhevsky 2009] (10/100 classes), and ResNet50 trained on ImageNet [Deng et al. 2009] (10000 classes). All 3 network architectures on CIFAR10/100 were trained 100 times (i.e., \( S = 100 \)) following the Deep Ensemble (DE) workflow, while ResNet50 was trained on ImageNet resulting in 50 models either by Deep Ensemble or by Snapshot Ensemble (i.e., \( S = 50 \)). For every sample in the standard testing dataset, \( S \) ensemble members were generated from the \( S \) models. While looking for the truth vector, for both the Vanilla Truth Discovery Ensemble (TDE) and its accuracy-preserving counterpart (aTDE), we set \( \epsilon = e^{-8} \) in all experiments, and observed a convergence within typically 5 iterations.

Results. The comparison of truth discovery ensemble methods with Deep Ensemble on CIFARs (\( S = 50 \) or 100), and with Deep Ensemble/Snapshot Ensemble on ImageNet (\( S = 25 \) or 50) is shown in Table 1. Clearly, TDE ameliorates either ECE or ECE\(_{KDE}\) by a large margin in most of the experimental settings, especially on datasets with higher complexity (i.e., CIFAR100 and ImageNet), but fails at maintain accuracy. The accuracy-preserving version aTDE, on the other hand, successfully preserves the accuracy, with nearly the same capability of lowering the ECEs, which validates the correctness of our accuracy-preserving Algorithm 2. It can also be concluded from Table 1 that higher ACC and lower ECEs are hard to be reached simultaneously, but the metrics contributed to by the two variants TDE/aTDE are usually very similar. The KS metric recently proposed by Gupta et al. [2021] which measures the maximal distance between the accumulated output probability to the actual probability, is a binning-free calibration evaluator that different from ECEs. The KS error is also measured for all the experiments. By leveraging truth discovery, our TDE method lowers the KS to as low as 0.6% (100 sources) as shown in Figure 2(e), even without any hold-out calibration sample.

Further, we investigate the stability of TDE/aTDE with different number of sources by changing \( S \) by an interval of 10 for CIFARs and 5 for ImageNet, as illustrated in Figure 2 and Tables A1, A2, A3 and A4. Interestingly, Deep Ensemble tends to be overconfident with larger number of sources, i.e., ensemble members, while TDE/aTDE works favorably with even larger amount of available sources, and this is when a high accuracy is usually reached (Figure 2), suggesting TDE and aTDE’s superior ability in utilizing information from multiple sources than Deep Ensemble.

5.2 IMPROVED POST-HOC CALIBRATION BY TRUTH DISCOVERY-REGULARIZED OPTIMIZATION

Experimental Setup. In this section, we evaluate the performance of our truth discovery-regularized post-hoc calibration methods (pTDE), to which the information elicited from ensemble-based methods is incorporated. To this end, we train ResNet18, DenseNet121 [Huang et al. 2017b] and ResNeXt29 [Xie et al. 2017] on CIFAR10/100 datasets with Snapshot Ensemble scheme and obtain 200 ensemble members as the sources (i.e., \( S = 200 \)). See Section A.2 for details. The pre-trained ImageNet models are also used with source numbers set at 10 and 30 for both DE and SE. We first train the vanilla optimization method using histogram-based ECE as the loss function (opt\(_{hist}\)) as described in Section 4.4 with batch size at 1000 for CIFARs and 10000 for ImageNet for 70 epochs. Then, we apply compositional training by switching to KDE-based loss function (opt\(_{KDE}\)) for 5 additional epochs. To leverage the information gained from truth discovery, the entropy based geometric variance (HV) values are computed for all \((N_c + N_e)\) samples. By taking HV into the training process, opt\(_{hist}\) is promoted to truth discovery-regularized post-hoc calibration pTDE\(_{hist}\), which is subsequently optimized for 5 additional epochs using ECE\(_{KDE}\) as the loss function to be pTDE\(_{KDE}\). Finally, pTDE\(_{KDE}\) (or pTDE for short) is compared with several state-of-the-art post-hoc calibration methods, namely, Temperature Scaling (TS) [Guo et al. 2017], Ensemble Temperature Scaling (ETS) [Zhang et al. 2020], and multi-class isotonic regression (IRM) [Zhang et al. 2020].

Results. The attenuation/enhancement factor we apply in Eq. 11 enables an iterative rearrangement of samples across bins, until a small discrepancy between confidence and actual accuracy is achieved within every bin. Figure 3 shows how this recalibration (vanilla opt\(_{hist}\)) affects confidence distribution. From Table 2 we can see that our method (pTDE) significantly outperforms all competing methods (except on CIFAR10 with ECE). It is worth noting that when pTDE is excluded, there is no sweeping method under every circumstances, but pTDE overall shows better consistency especially with the ECE\(_{KDE}\) metric which is not susceptible to the binning strategy, e.g., the number and positions of the bins. To inspect the individual contributions

   "downloaded from https://github.com/bayesgroup/pytorch-ensembles"
of compositional training and truth discovery regularization to the total performance, we conduct ablation study for the two components as shown in Table 3. Generally, the truth discovery-regularized methods (pTDE\textsuperscript{hist} and pTDE\textsuperscript{KDE}) perform better. Depending on which kind of calibration error is used as loss function, it is purposefully minimized. This indicates that pTDE\textsuperscript{hist} performs better with ECE metric, while pTDE\textsuperscript{KDE} favors ECE\textsuperscript{KDE} metric. Notably, although ECE\textsuperscript{KDE} is targeted by pTDE\textsuperscript{KDE}, ECE can still be decreased on ImageNet, for example, when 30 sources from Deep Ensemble is used, showing the effectiveness of our proposed compositional training and truth discovery regularization.

Finally, all the calibration results are further measured upon KS, shown in Table 4, and surprisingly, although pTDE is not specifically designed for the optimization of KS, its fully-fledged version pTDE\textsuperscript{KDE} is competitive, or even better than Spline, showing that the truth discovery-based regularizer is also beneficial to the minimization of the KS metric.

### 6 CONCLUSION

In this work, we first present Truth Discovery Ensemble (TDE) that neither requires hold-out calibration data nor alters any training process, but significantly surpasses the original ensemble result, and in the meanwhile preserves the accuracy (aTDE). For post-hoc calibration, the superiority of our final methods (pTDE) is attributed not only to truth discovery, but also to the compositional training strategy. In conclusion, truth discovery is well positioned to assist both ensemble-based and post-hoc calibration. We hope that the proposed calibrators augmented by truth discovery can enlarge the arsenal of uncertainty calibration methods for deep learning. Our source code is available at https://github.com/horsepurve/truly-uncertain.

### Acknowledgements

This research was supported in part by NSF through grants CCF-1716400 and IIS-1910492.

### References

Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. In International Conference on Learning Representations, 2020. URL https://openreview.net/forum?id=BJxI5gHKDr

Yu Bai, Tengyu Ma, Huan Wang, and Caiming Xiong. Improved uncertainty post-calibration via rank preserving transforms, 2021. URL https://openreview.net/forum?id=jsM6yvqlT0W
Raef Bassily, Adam Groce, Jonathan Katz, and Adam D. Smith. Coupled-worlds privacy: Exploiting adversarial uncertainty in statistical data privacy. In 54th Annual IEEE Symposium on Foundations of Computer Science, FOCS, pages 439–448, 2013.

Jeffrey De Fauw, Joseph R Ledsam, Bernardino Romera-Paredes, Stanislav Nikolov, Nenad Tomasev, Sam Blackwell, Harry Askham, Xavier Glorot, Brendan O'Donoghue, Daniel Visentin, et al. Clinically applicable deep learning for diagnosis and referral in retinal disease. *Nature Medicine*, 24(9):1342–1350, 2018.

Jia Deng,Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pages 248–255, 2009.

Hu Ding and Jinhui Xu. Learning the truth vector in high dimensions. *Journal of Computer and System Sciences*, pages 78–94, 2020.

Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, 2017.

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

Yarin Gal, Jiri Hron, and Alex Kendall. Concrete Dropout. In *Advances in Neural Information Processing Systems 30 (NIPS)*, 2017.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P. Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8803–8812, 2018.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

Kartik Gupta, Amir Rahimi, Thalaiyasingam Ajanthan, Thomas Mensink, Cristian Sminchisescu, and Richard Hartley. Calibration of neural networks using splines. In *International Conference on Learning Representations*, 2021. URL [https://openreview.net/forum?id=eQe8DEWNN2W](https://openreview.net/forum?id=eQe8DEWNN2W)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In 2016 *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 770–778, 2016.

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning*, pages 2712–2721. PMLR, 2019a.

Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *Advances in Neural Information Processing Systems*, pages 15637–15648, 2019b.

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E. Hopcroft, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 2261–2269, 2017b.

Ziyun Huang, Hu Ding, and Jinhui Xu. A faster algorithm for truth discovery via range cover. *Algorithmica*, 81(10):4118–4133, 2019.

Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 66–75, 2017.

Alex Krizhevsky. Learning multiple layers of features from tiny images. *Technical report*, 2009.

Meelis Kull, Miquel Perello Nieto, Markus Küngsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In *Advances in Neural Information Processing Systems*, pages 12295–12305, 2019.

Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, Proceedings of Machine Learning Research, pages 2810–2819, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6405–6416, 2017.

Qi Li, Yaliang Li, Jing Gao, Bo Zhao, Wei Fan, and Jiawei Han. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *International Conference on Management of Data, SIGMOD*, pages 1187–1198, 2014.
Shi Li, Jinhui Xu, and Minwei Ye. Approximating global optimum for probabilistic truth discovery. *Algorithmica*, pages 3091–3116, 2020.

Chunwei Ma, Yan Ren, Jiarui Yang, Zhe Ren, Huanming Yang, and Siqi Liu. Improved peptide retention time prediction in liquid chromatography through deep learning. *Analytical Chemistry*, 90(18):10881–10888, 2018.

Chunwei Ma, Zhanghexuan Ji, and Mingchen Gao. Neural style transfer improves 3D cardiovascular MR image segmentation on inconsistent data. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 128–136. Springer, 2019.

Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems*, pages 13132–13143, 2019.

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

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*, pages 2901–2907, 2015.

Travis A. O’Brien, Karthik Kashinath, Nicholas R. Cavanaugh, William D. Collins, and John P. O’Brien. A fast and objective multidimensional kernel density estimation method: fastkde. *Computational Statistics & Data Analysis*, 101:148–160, 2016.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8024–8035, 2019.

Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tammoj Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *Advances in Neural Information Processing Systems*, pages 13888–13899, 2019.

Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 5987–5995, 2017.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Richard C. Wilson, Edwin R. Hancock, and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference 2016, BMVC*, 2016.

Jize Zhang, Bhavya Kailkhura, and T. Yong-Jin Han. Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML*, volume 119, pages 11117–11128, 2020.
We solve the above optimization problem using Lagrange Multipliers with Karush-Kuhn-Tucker (KKT) Conditions. Let
\[ z_l < c \quad \forall l \neq c \]
\[ 0 \leq z_l' \leq 1 \quad \forall l' \]
\[ \sum_{l'=1}^L z_l' = 1. \]

This projection can be found through the following constrained optimization problem:
\[
\begin{align*}
\text{minimize} & \quad (z_1 - z_1^*)^2 + (z_2 - z_2^*)^2 + \ldots + (z_L - z_L^*)^2 \\
\text{s.t.} & \quad 0 \leq z_l \leq c \\
& \quad 0 \leq z_l' \leq 1 \\
& \quad \sum_{l'=1}^L z_l' = 1.
\end{align*}
\]

We solve the above optimization problem using Lagrange Multipliers with Karush-Kuhn-Tucker (KKT) Conditions. Let \( \lambda_0, \lambda_1, \ldots, \lambda_{c-1} \geq 0, \eta_1, \ldots, \eta_{c-1} \in \mathbb{R} \) and define
\[ G = \sum_{l'=1}^L (z_l' - z_l^*)^2 + \sum_{\forall l \neq c} \lambda_l (z_l - z_c + \eta_l^2) + \lambda_0 (z_1 + \ldots + z_L - 1). \]

Taking the derivative with respect to \( z \) and the multipliers gives
\[
\begin{align*}
\frac{\partial G}{\partial z_c} &= 2(z_c - z_c^*) + \lambda_0 + \lambda_1 + \ldots + \lambda_{c-1} \\
\frac{\partial G}{\partial z_l} &= 2(z_l^* - z_l) + \lambda_0 - \lambda_l \quad \forall l \neq c \\
\frac{\partial G}{\partial \lambda_l} &= z_l - z_c + \eta_l^2 \quad \forall l \neq c \\
\frac{\partial G}{\partial \eta_l} &= 2\lambda_l \eta_l \quad \forall l \neq c.
\end{align*}
\]

Setting the gradient of the Lagrangian \( G \) to 0 and rearranging gives us \( \lambda_l = 2(z_l^* - z_l), \quad z_l - z_c + \eta_l^2 = 0, \quad \text{and} \quad \lambda_l \eta_l = 0, \quad \forall l \neq c. \) For every \( \lambda_l \leq 0 \), there are two possibilities: (I) if \( \lambda_l = 0 \), then \( z_l = z_l^* \) which means that the \( l^{th} \) components will not be changed; (II) if \( \lambda_l > 0 \), then \( z_l \) will be decreased to be the same as \( z_c \). In Algorithm 2, we first sort the components of \( z^* \), denoted as \( \{z_l^*\}_{l=1}^L \), then the components less than \( z_c^* \) should be fixed, or otherwise it will contradict the above two conditions. Then, we iterate through the top \( \tilde{l} \) components of the sorted components until we find an \( \tilde{l} \) such that the following condition holds:
\[
\frac{1}{\tilde{l} + 1} (z_c^* + \sum_{l'=1}^{\tilde{l}} z_l'^*) > z_{\tilde{l}+1}^*.
\]

Since \( \{z_l^*\}_{l=1}^L \) is sorted, Inequal. 17 implies that all conditions in problem 16 hold. Therefore, the solution
\[
\tilde{z}_{l'} = \begin{cases} 
\frac{1}{\tilde{l} + 1} (z_c^* + \sum_{l'=1}^{\tilde{l}} z_l'^*) & \forall l' \leq \tilde{l} \\
\tilde{z}_l' & \forall l' > \tilde{l}
\end{cases}
\]
given by Algorithm 3.1 will not alter the index of the maximum component.

A.2 EXPERIMENTAL DETAILS

In this section we describe the detailed experimental setups for post-hoc calibration. Three architectures ResNet18, DenseNet121, and ResNeXt29 are trained on CIFAR10/100 for 200 epochs with a batch size of 128. The learning rate...
is 0.1 initially and multiplied by 0.2 at the 60th, 120th, and 160th epoch. For all experiment SGD optimizer is used with momentum at 0.9 and weight decay at 5x-4. To obtain as many sources as possible in reasonable time, we treat all models from 200 snapshots as sources for truth discovery and for the computation of the Entropy based Geometric Variance (HV). In the post-hoc calibration step, all the original testing sets are further randomly split into two equal-sized calibration/evaluation datasets, i.e., \( N_c = N_e = 5000 \) for CIFAR10/100 and \( N_c = N_e = 25000 \) for ImageNet. This random split is performed 5 times using different seeds, and the resulting means and standard deviations are reported. For all the experiments, the number of bins \( B \) is set at 15 and their endpoints are determined by evenly distributing all calibration samples into the 15 bins. Then the number and locations of the bins are fixed during the optimizations. As for the hyperparameters in Eq. \([15]\), we simply set \( \alpha_1 = 1, \alpha_2 = \psi_h \) without manually tuning, in order to keep minimal human intervention.

### A.3 ADDITIONAL TABLES & FIGURES

| #sources | CIFAR100 | CIFAR100 |
|---|---|---|
| | ECE | ECE | NLL | MSE | KS | ACC | ECE | ECE | NLL | MSE | KS | ACC |
| DE | 20 | 2.22 | 2.10 | 0.638 | 0.252 | 1.61 | 82.32 | 1.10 | 0.50 | 0.113 | 0.054 | 0.19 | 96.28 |
| TDE | 20 | 2.27 | 3.06 | 0.648 | 0.254 | 2.78 | 82.21 | 1.38 | 1.35 | 0.119 | 0.056 | 1.35 | 96.32 |
| aTDE | 20 | 2.37 | 3.16 | 0.648 | 0.254 | 2.63 | 82.32 | 1.43 | 1.38 | 0.119 | 0.056 | 1.38 | 96.28 |
| DE | 30 | 2.65 | 2.43 | 0.618 | 0.247 | 2.07 | 82.67 | 1.25 | 0.45 | 0.110 | 0.053 | 0.23 | 96.42 |
| TDE | 30 | 1.68 | 1.90 | 0.621 | 0.247 | 1.48 | 82.61 | 1.23 | 1.00 | 0.112 | 0.054 | 0.39 | 96.39 |
| aTDE | 30 | 1.66 | 1.94 | 0.620 | 0.247 | 1.41 | 82.67 | 1.21 | 0.97 | 0.112 | 0.054 | 0.96 | 96.42 |
| DE | 40 | 2.93 | 2.69 | 0.610 | 0.245 | 2.33 | 82.92 | 1.27 | 0.43 | 0.107 | 0.052 | 0.25 | 96.42 |
| TDE | 40 | 1.59 | 1.92 | 0.612 | 0.245 | 0.85 | 82.87 | 1.15 | 0.83 | 0.109 | 0.053 | 0.82 | 96.42 |
| aTDE | 40 | 1.61 | 1.97 | 0.612 | 0.245 | 0.80 | 82.92 | 1.19 | 0.83 | 0.109 | 0.053 | 0.82 | 96.42 |
| DE | 50 | 2.83 | 2.48 | 0.606 | 0.244 | 2.17 | 82.79 | 1.33 | 0.41 | 0.107 | 0.052 | 0.25 | 96.41 |
| TDE | 50 | 1.61 | 2.05 | 0.606 | 0.244 | 0.72 | 82.86 | 1.09 | 0.78 | 0.109 | 0.052 | 0.77 | 96.39 |
| aTDE | 50 | 1.54 | 1.98 | 0.606 | 0.244 | 0.71 | 82.79 | 1.07 | 0.76 | 0.109 | 0.052 | 0.75 | 96.41 |
| DE | 60 | 2.89 | 2.50 | 0.602 | 0.244 | 2.20 | 82.83 | 1.31 | 0.48 | 0.106 | 0.052 | 0.23 | 96.38 |
| TDE | 60 | 1.61 | 1.99 | 0.602 | 0.243 | 0.78 | 82.89 | 1.06 | 0.76 | 0.107 | 0.052 | 0.75 | 96.38 |
| aTDE | 60 | 1.55 | 1.93 | 0.602 | 0.243 | 0.72 | 82.83 | 1.07 | 0.76 | 0.107 | 0.052 | 0.75 | 96.38 |
| DE | 70 | 3.06 | 2.62 | 0.600 | 0.243 | 2.39 | 83.00 | 1.31 | 0.38 | 0.106 | 0.052 | 0.23 | 96.45 |
| TDE | 70 | 1.75 | 1.98 | 0.600 | 0.242 | 0.92 | 82.96 | 1.05 | 0.71 | 0.107 | 0.052 | 0.69 | 96.40 |
| aTDE | 70 | 1.79 | 2.02 | 0.600 | 0.242 | 0.96 | 83.00 | 1.09 | 0.66 | 0.107 | 0.052 | 0.64 | 96.45 |
| DE | 80 | 3.06 | 2.63 | 0.598 | 0.243 | 2.41 | 83.02 | 1.24 | 0.39 | 0.106 | 0.052 | 0.21 | 96.42 |
| TDE | 80 | 1.85 | 2.02 | 0.599 | 0.242 | 1.03 | 83.04 | 1.04 | 0.70 | 0.107 | 0.052 | 0.68 | 96.38 |
| aTDE | 80 | 1.82 | 2.00 | 0.599 | 0.242 | 1.01 | 83.02 | 1.04 | 0.66 | 0.107 | 0.052 | 0.64 | 96.42 |
| DE | 90 | 3.06 | 2.63 | 0.596 | 0.243 | 2.38 | 83.00 | 1.21 | 0.39 | 0.105 | 0.052 | 0.25 | 96.46 |
| TDE | 90 | 1.76 | 2.00 | 0.597 | 0.242 | 1.01 | 83.00 | 1.03 | 0.59 | 0.106 | 0.052 | 0.59 | 96.45 |
| aTDE | 90 | 1.78 | 2.00 | 0.597 | 0.242 | 1.01 | 83.00 | 1.01 | 0.58 | 0.106 | 0.052 | 0.58 | 96.46 |
| DE | 100 | 3.06 | 2.59 | 0.596 | 0.243 | 2.39 | 83.03 | 1.18 | 0.39 | 0.105 | 0.052 | 0.22 | 96.44 |
| TDE | 100 | 1.83 | 2.03 | 0.596 | 0.242 | 1.10 | 83.06 | 0.99 | 0.57 | 0.106 | 0.052 | 0.57 | 96.46 |
| aTDE | 100 | 1.80 | 2.00 | 0.596 | 0.242 | 1.07 | 83.03 | 0.98 | 0.59 | 0.106 | 0.052 | 0.59 | 96.44 |

Table A1: Comparison between DE/TDE/aTDE on PreResNet110. The best results are highlighted.
Table A2: Comparison between DE/TDE/aTDE on PreResNet164. The best results are highlighted.

| #sources | CIFAR100 | CIFAR10 |
|----------|----------|---------|
|          | $ECE_{\downarrow}$ | $ECE_{\downarrow}$ | $NLL_{\downarrow}$ | $MSE_{\downarrow}$ | $KS_{\downarrow}$ | $ACC_{\uparrow}$ | $ECE_{\downarrow}$ | $ECE_{\downarrow}$ | $NLL_{\downarrow}$ | $MSE_{\downarrow}$ | $KS_{\downarrow}$ | $ACC_{\uparrow}$ |
| DE 10 | 1.90 | 1.86 | 0.618 | 0.242 | 1.00 | 82.95 | 0.93 | 0.32 | 0.108 | 0.052 | 0.17 | 96.57 |
| TDE 20 | 1.73 | 2.45 | 0.602 | 0.239 | 1.86 | 83.30 | 1.24 | 0.10 | 0.115 | 0.053 | 1.22 | 96.57 |
| aTDE 30 | 1.67 | 2.46 | 0.602 | 0.239 | 1.86 | 83.30 | 1.24 | 0.10 | 0.115 | 0.053 | 1.22 | 96.57 |
| DE 40 | 1.90 | 1.90 | 0.618 | 0.242 | 1.00 | 82.95 | 0.93 | 0.32 | 0.108 | 0.052 | 0.17 | 96.57 |
| TDE 50 | 1.73 | 2.45 | 0.602 | 0.239 | 1.86 | 83.30 | 1.24 | 0.10 | 0.115 | 0.053 | 1.22 | 96.57 |
| aTDE 60 | 1.67 | 2.46 | 0.602 | 0.239 | 1.86 | 83.30 | 1.24 | 0.10 | 0.115 | 0.053 | 1.22 | 96.57 |
| DE 70 | 1.90 | 1.90 | 0.618 | 0.242 | 1.00 | 82.95 | 0.93 | 0.32 | 0.108 | 0.052 | 0.17 | 96.57 |
| TDE 80 | 1.73 | 2.45 | 0.602 | 0.239 | 1.86 | 83.30 | 1.24 | 0.10 | 0.115 | 0.053 | 1.22 | 96.57 |
| aTDE 90 | 1.67 | 2.46 | 0.602 | 0.239 | 1.86 | 83.30 | 1.24 | 0.10 | 0.115 | 0.053 | 1.22 | 96.57 |
| DE 100 | 1.90 | 1.90 | 0.618 | 0.242 | 1.00 | 82.95 | 0.93 | 0.32 | 0.108 | 0.052 | 0.17 | 96.57 |
| TDE 110 | 1.73 | 2.45 | 0.602 | 0.239 | 1.86 | 83.30 | 1.24 | 0.10 | 0.115 | 0.053 | 1.22 | 96.57 |
| aTDE 120 | 1.67 | 2.46 | 0.602 | 0.239 | 1.86 | 83.30 | 1.24 | 0.10 | 0.115 | 0.053 | 1.22 | 96.57 |
Table A3: Comparison between DE/TDE/aTDE on WideResNet28x10. The best results are highlighted.

| #sources | CIFAR100 | | | | | CIFAR10 | | | |
|----------|---------|----------|----------|----------|----------|---------|----------|----------|----------|
|          | ECE,KDE↓ | ECE↓ | NLL↓ | MSE↓ | KS↓ | ACC↑ | ECE,KDE↓ | ECE↓ | NLL↓ | MSE↓ | KS↓ | ACC↑ |
| DE 10    | 6.00     | 5.41   | 0.629 | 0.235 | 5.22 | 83.97 | 1.04     | 0.24   | 0.094 | 0.045 | 0.21 | 96.93 |
| TDE 10   | 4.73     | 4.69   | 0.641 | 0.237 | 4.01 | 84.10 | 1.28     | 1.06   | 0.099 | 0.047 | 1.05 | 96.93 |
| aTDE 10  | 4.61     | 4.59   | 0.641 | 0.237 | 3.91 | 83.97 | 1.28     | 1.06   | 0.099 | 0.047 | 1.04 | 96.93 |
| DE 20    | 6.27     | 5.70   | 0.621 | 0.233 | 5.49 | 84.21 | 0.96     | 0.23   | 0.092 | 0.045 | 0.11 | 97.02 |
| TDE 20   | 5.05     | 4.62   | 0.628 | 0.234 | 4.25 | 84.23 | 1.19     | 0.80   | 0.096 | 0.046 | 0.78 | 97.00 |
| aTDE 20  | 5.04     | 4.61   | 0.628 | 0.234 | 4.24 | 84.23 | 1.18     | 0.77   | 0.096 | 0.046 | 0.76 | 97.02 |
| DE 30    | 6.28     | 5.73   | 0.617 | 0.232 | 5.49 | 84.22 | 1.07     | 0.33   | 0.091 | 0.045 | 0.24 | 97.15 |
| TDE 30   | 5.20     | 4.69   | 0.623 | 0.233 | 4.35 | 84.26 | 1.15     | 0.60   | 0.094 | 0.045 | 0.57 | 97.12 |
| aTDE 30  | 5.16     | 4.66   | 0.623 | 0.233 | 4.32 | 84.22 | 1.17     | 0.57   | 0.094 | 0.045 | 0.54 | 97.15 |
| DE 40    | 6.39     | 5.75   | 0.616 | 0.232 | 5.58 | 84.28 | 1.09     | 0.28   | 0.091 | 0.044 | 0.21 | 97.14 |
| TDE 40   | 5.39     | 4.86   | 0.622 | 0.232 | 4.54 | 84.31 | 1.22     | 0.53   | 0.093 | 0.045 | 0.57 | 97.14 |
| aTDE 40  | 5.36     | 4.83   | 0.622 | 0.232 | 4.51 | 84.28 | 1.27     | 0.53   | 0.093 | 0.045 | 0.51 | 97.14 |
| DE 50    | 6.45     | 5.86   | 0.615 | 0.232 | 5.64 | 84.38 | 1.07     | 0.35   | 0.091 | 0.044 | 0.24 | 97.17 |
| TDE 50   | 5.48     | 5.04   | 0.621 | 0.232 | 4.54 | 84.37 | 1.15     | 0.49   | 0.093 | 0.045 | 0.47 | 97.15 |
| aTDE 50  | 5.50     | 5.05   | 0.621 | 0.232 | 4.63 | 84.38 | 1.14     | 0.47   | 0.093 | 0.045 | 0.45 | 97.17 |
| DE 60    | 6.39     | 5.72   | 0.614 | 0.232 | 5.57 | 84.30 | 1.05     | 0.34   | 0.091 | 0.044 | 0.21 | 97.14 |
| TDE 60   | 5.53     | 5.00   | 0.619 | 0.232 | 4.66 | 84.36 | 1.14     | 0.49   | 0.093 | 0.045 | 0.47 | 97.13 |
| aTDE 60  | 5.47     | 4.94   | 0.619 | 0.232 | 4.67 | 84.31 | 1.13     | 0.48   | 0.093 | 0.045 | 0.46 | 97.14 |
| DE 70    | 6.42     | 5.75   | 0.614 | 0.232 | 5.59 | 84.31 | 1.10     | 0.32   | 0.090 | 0.044 | 0.23 | 97.13 |
| TDE 70   | 5.55     | 5.02   | 0.619 | 0.232 | 4.70 | 84.34 | 1.10     | 0.46   | 0.092 | 0.045 | 0.44 | 97.14 |
| aTDE 70  | 5.52     | 4.99   | 0.619 | 0.232 | 4.67 | 84.31 | 1.11     | 0.45   | 0.092 | 0.045 | 0.43 | 97.15 |
| DE 80    | 6.44     | 5.80   | 0.615 | 0.232 | 5.62 | 84.30 | 1.14     | 0.33   | 0.090 | 0.044 | 0.26 | 97.16 |
| TDE 80   | 5.54     | 5.00   | 0.619 | 0.232 | 4.70 | 84.27 | 1.10     | 0.45   | 0.092 | 0.044 | 0.41 | 97.15 |
| aTDE 80  | 5.57     | 5.03   | 0.619 | 0.232 | 4.74 | 84.30 | 1.11     | 0.44   | 0.092 | 0.044 | 0.40 | 97.16 |
| DE 90    | 6.43     | 5.83   | 0.614 | 0.232 | 5.63 | 84.30 | 1.20     | 0.42   | 0.090 | 0.044 | 0.29 | 97.19 |
| TDE 90   | 5.59     | 5.11   | 0.619 | 0.232 | 4.75 | 84.31 | 1.12     | 0.39   | 0.092 | 0.044 | 0.36 | 97.19 |
| aTDE 90  | 5.58     | 5.10   | 0.619 | 0.232 | 4.74 | 84.30 | 1.12     | 0.39   | 0.092 | 0.044 | 0.36 | 97.19 |
| DE 100   | 6.41     | 5.79   | 0.614 | 0.231 | 5.61 | 84.28 | 1.20     | 0.41   | 0.090 | 0.044 | 0.30 | 97.20 |
| TDE 100  | 5.58     | 5.08   | 0.618 | 0.232 | 4.74 | 84.29 | 1.11     | 0.40   | 0.092 | 0.044 | 0.36 | 97.17 |
| aTDE 100 | 5.57     | 5.07   | 0.618 | 0.232 | 4.73 | 84.28 | 1.13     | 0.42   | 0.092 | 0.044 | 0.33 | 97.20 |
Table A4: Comparison between DE/TDE/aTDE on ImageNet. The best results are highlighted.

| #sources | ImageNet | ImageNet (Snapshot Ensemble) |
|----------|----------|-----------------------------|
|          | ECE ▼ | ECE ▼ | NLL ▼ | MSE ▼ | KS ▼ | ACC ▼ | ECE ▼ | ECE ▼ | NLL ▼ | MSE ▼ | KS ▼ | ACC ▼ |
| DE       | 2.32 | 2.45 | 0.831 | 0.302 | 2.04 | 78.65 | 2.00 | 2.18 | 0.866 | 0.313 | 2.18 | 77.79 |
| TDE      | 1.99 | 2.43 | 0.838 | 0.304 | 0.90 | 78.38 | 1.99 | 2.46 | 0.874 | 0.315 | 2.46 | 77.61 |
| aTDE     | 2.33 | 2.78 | 0.837 | 0.304 | 1.26 | 78.65 | 2.21 | 2.69 | 0.873 | 0.315 | 2.69 | 77.79 |
| DE       | 2.94 | 2.94 | 0.808 | 0.297 | 2.60 | 79.15 | 2.12 | 2.32 | 0.843 | 0.307 | 2.32 | 78.25 |
| TDE      | 1.96 | 2.31 | 0.809 | 0.297 | 1.41 | 79.02 | 1.78 | 2.17 | 0.846 | 0.307 | 2.17 | 78.17 |
| aTDE     | 2.11 | 2.46 | 0.809 | 0.297 | 1.56 | 79.15 | 1.88 | 2.25 | 0.846 | 0.307 | 2.26 | 78.25 |
| DE       | 2.96 | 2.92 | 0.802 | 0.295 | 2.61 | 79.15 | 1.98 | 2.20 | 0.837 | 0.304 | 2.20 | 78.37 |
| TDE      | 2.04 | 2.32 | 0.801 | 0.295 | 1.60 | 79.17 | 1.74 | 2.06 | 0.839 | 0.305 | 2.06 | 78.32 |
| aTDE     | 2.03 | 2.31 | 0.801 | 0.295 | 1.59 | 79.15 | 1.79 | 2.12 | 0.839 | 0.305 | 2.12 | 78.37 |
| DE       | 3.10 | 3.07 | 0.799 | 0.295 | 2.76 | 79.26 | 1.89 | 2.14 | 0.834 | 0.304 | 2.14 | 78.42 |
| TDE      | 2.14 | 2.36 | 0.798 | 0.294 | 1.75 | 79.22 | 1.65 | 2.04 | 0.837 | 0.304 | 2.04 | 78.35 |
| aTDE     | 2.20 | 2.42 | 0.798 | 0.294 | 1.81 | 79.26 | 1.73 | 2.08 | 0.837 | 0.304 | 2.08 | 78.42 |
| DE       | 3.11 | 3.12 | 0.797 | 0.295 | 2.79 | 79.25 | 1.86 | 2.12 | 0.833 | 0.303 | 2.12 | 78.46 |
| TDE      | 2.17 | 2.42 | 0.796 | 0.294 | 1.81 | 79.22 | 1.69 | 2.09 | 0.836 | 0.303 | 2.09 | 78.55 |
| aTDE     | 2.23 | 2.47 | 0.794 | 0.293 | 1.88 | 79.27 | 1.75 | 2.11 | 0.836 | 0.303 | 2.11 | 78.45 |
| DE       | 3.11 | 3.09 | 0.795 | 0.294 | 2.78 | 79.27 | 1.82 | 2.08 | 0.832 | 0.303 | 2.08 | 78.48 |
| TDE      | 2.24 | 2.47 | 0.794 | 0.293 | 1.88 | 79.26 | 1.72 | 2.11 | 0.835 | 0.303 | 2.11 | 78.45 |
| aTDE     | 2.29 | 2.50 | 0.793 | 0.293 | 1.84 | 79.27 | 1.77 | 2.12 | 0.836 | 0.303 | 2.12 | 78.46 |
| DE       | 3.14 | 3.10 | 0.794 | 0.294 | 2.80 | 79.27 | 1.79 | 2.07 | 0.831 | 0.303 | 2.07 | 78.50 |
| TDE      | 2.28 | 2.49 | 0.793 | 0.293 | 1.92 | 79.26 | 1.73 | 2.14 | 0.834 | 0.304 | 2.14 | 78.50 |
| aTDE     | 2.29 | 2.50 | 0.793 | 0.293 | 1.94 | 79.27 | 1.74 | 2.15 | 0.834 | 0.303 | 2.15 | 78.50 |
| DE       | 3.17 | 3.12 | 0.794 | 0.294 | 2.82 | 79.30 | 1.77 | 2.06 | 0.830 | 0.302 | 2.06 | 78.51 |
| TDE      | 2.31 | 2.50 | 0.793 | 0.293 | 1.97 | 79.29 | 1.72 | 2.13 | 0.833 | 0.302 | 2.13 | 78.51 |
| aTDE     | 2.33 | 2.52 | 0.793 | 0.293 | 1.98 | 79.30 | 1.72 | 2.14 | 0.833 | 0.302 | 2.14 | 78.51 |
| DE       | 3.21 | 3.14 | 0.793 | 0.294 | 2.86 | 79.34 | 1.78 | 2.06 | 0.830 | 0.302 | 2.06 | 78.51 |
| TDE      | 2.35 | 2.54 | 0.792 | 0.293 | 2.01 | 79.31 | 1.72 | 2.18 | 0.833 | 0.302 | 2.18 | 78.51 |
| aTDE     | 2.38 | 2.55 | 0.792 | 0.292 | 2.04 | 79.34 | 1.74 | 2.20 | 0.833 | 0.302 | 2.20 | 78.54 |
| DE       | 3.24 | 3.17 | 0.792 | 0.294 | 2.89 | 79.37 | 1.73 | 2.04 | 0.830 | 0.302 | 2.04 | 78.52 |
| TDE      | 2.41 | 2.58 | 0.791 | 0.293 | 2.07 | 79.35 | 1.66 | 2.11 | 0.833 | 0.302 | 2.11 | 78.50 |
| aTDE     | 2.44 | 2.61 | 0.791 | 0.293 | 2.10 | 79.37 | 1.69 | 2.14 | 0.833 | 0.302 | 2.14 | 78.52 |
Figure A1: Correlation between Entropy based Geometric Variance (HV) and winning score of DenseNet121 trained on CIFAR10. Winning score stands for the score of predicted label, i.e. the maximum score of the network’s output.

Figure A2: Correlation between Entropy based Geometric Variance (HV) and winning score of ResNet18 trained on CIFAR100.

Figure A3: Correlation of Entropy based Geometric Variance (HV) and winning score of ResNet50 trained on ImageNet using DE.