A Geometric Perspective towards Neural Calibration via Sensitivity Decomposition

Junjiao Tian
Georgia Institute of Technology
jtian73@gatech.edu

Dylan Yung
Georgia Institute of Technology
dyung6@gatech.edu

Yen-Chang Hsu
Samsung Research America
yenchang.hsu@samsung.com

Zsolt Kira
Georgia Institute of Technology
zkira@gatech.edu

Abstract

It is well known that vision classification models suffer from poor calibration in the face of data distribution shifts. In this paper, we take a geometric approach to this problem. We propose *Geometric Sensitivity Decomposition (GSD)* which decomposes the norm of a sample feature embedding and the angular similarity to a target classifier into an *instance-dependent* and an *instance-independent* component. The instance-dependent component captures the sensitive information about changes in the input while the instance-independent component represents the insensitive information serving solely to minimize the loss on the training dataset. Inspired by the decomposition, we analytically derive a simple extension to current softmax-linear models, which learns to disentangle the two components during training. On several common vision models, the disentangled model outperforms other calibration methods on standard calibration metrics in the face of out-of-distribution (OOD) data and corruption with significantly less complexity. Specifically, we surpass the current state of the art by 30.8% relative improvement on corrupted CIFAR100 in Expected Calibration Error. Code available at https://github.com/GT-RIPL/Geometric-Sensitivity-Decomposition.git.

1 Introduction

During development, deep learning models are trained and validated on data from the same distribution. However, in the real world sensors degrade and weather conditions change. Similarly, subtle changes in image acquisition and processing can also lead to distribution shift of the input data. This is often known as *covariate shift*, and will typically decrease the performance (e.g. classification accuracy). However, it has been empirically found that the model's *confidence* remains high even when accuracy has degraded [1]. The process of aligning confidence to empirical accuracy is called model *calibration*. Calibrated probability provides valuable uncertainty information for decision making. For example, knowing when a decision cannot be trusted and more data is needed is important for safety and efficiency in real world applications such as self-driving [2] and active learning [3].

A comprehensive comparison of calibration methods has been studied for in-distribution (IND) data [4], However, these methods lead to unsatisfactory performance under distribution shift [5]. To resolve the problem, high-quality uncertainty estimation [6, 5] is required. Principled Bayesian methods [7] model uncertainty directly but are computationally heavy. Recent deterministic methods [8, 9] propose to improve a model's *sensitivity* to input changes by regularizing the model's intermediate layers. In this context, sensitivity is defined as preserving distance between two different input samples through layers of the model. We would like to utlize the improved sensitivity to better detect

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

Out-of-Distribution (OOD) data. However, these methods introduce added architecture changes and large combinatorics of hyperparameters.

Unlike existing works, we propose to study sensitivity from a geometric perspective. The last linear layer in a softmax-linear model can be decomposed into the multiplication of a norm and a cosine similarity term [10, 11, 12, 13]. Geometrically, the angular similarity dictates the membership of an input and the norm only affects the confidence in a softmax-linear model. Counter-intuitively, the norm of a sample's feature embedding exhibits little correlation to the hardness of the input [11]. Based on this observation, we explore two questions: 1) why is a model's confidence insensitive to distribution shift? 2) how do we improve model sensitivity and calibration?

We hypothesize that in part an insensitive norm is responsible for bad calibration especially on shifted data. We observe that the sensitivity of the angular similarity increases with training whereas the sensitivity of the norm remains low. More importantly, calibration worsens during the period when the norm increases while the angular similarity changes slowly. This shows a concrete example of the inability of the norm to *adapt* when accuracy has dropped. Intuitively, training on clean datasets encourages neural networks to *always* output increasingly large feature norm to continuously minimize the training loss. Because the probability of the prevalent class of an input is proportional to its norm, larger norms lead to smaller training loss when most training data have been classified correctly (See Sec. 3.1). This renders the norm insensitive to input differences because the model is trained to *always* output features with large norm on clean data. While we have put forth that the norm is poorly calibrated, we must emphasize that it can still play an important role in model calibration (See Sec. 4.1).

To encourage sensitivity, we propose to decompose the norm of a sample's feature embedding and the angular similarity into two components: *instance-dependent* and *instance-independent*. The instance-dependent component captures the sensitive information about the input while the instance-independent component represents the insensitive information serving solely to minimize the loss on the training dataset. Inspired by the decomposition, we analytically derive a simple extension to the current softmax-linear model, which learns to disentangle the two components during training. We show that our model outperforms other deterministic methods (despite their significant complexity) and is comparable to multi-pass methods with fewer training hyperparameters in Sec. 4.1.

In summary, our contributions are four fold:

- We study the problem of calibration geometrically and identify that the insensitive norm is responsible for bad calibration under distribution shift.
- We derive a principled but simple geometric decomposition that decomposes the norm into an instance-dependent and instance-independent component.
- Based on the decomposition, we propose a simple training and inference scheme to encourage the norm to reflect distribution changes.
- We achieve state of the art results in calibration metrics in the face of corruptions while having arguably the simplest calibration method to implement.

2 Related Work

Methods dedicated to strengthening calibration can be divided into two camps: multi-pass models and single-pass deterministic models. The current state-of-the-art multi-pass models are: Bayesian Monte Carlo Drop Out (MCDO) [7] and Deep Ensembles [14]. Bayesian methods are the most principled way to model uncertainty. Instead of *optimizing* max likelihood for a single set of parameters, Bayesian methods obtain a posterior distribution over possible parameters given a prior distribution over parameters and calculated data likelihood assuming some process noise. The posterior distribution over parameters captures epistemic uncertainty or uncertainty due to the limits of what the model knows . The final predictive distribution is obtained by *marginalizing* out model parameters. While Bayesian methods are theoretically sound, they are intractable in practice. Deep Ensembles leverage multiple models trained using different random initialization of weights so models learn different classification functions, and these variations then ensembled by averaging their predicted probabilities.

A recent trend is to use a single-pass deterministic **non-Bayesian** model to improve uncertainty estimation. Two recent works DUQ [8] and SNGP [9] propose to improve uncertainty-awarenesss of deterministic networks by improving the networks' sensitivity to input changes. Intuitively, a sensitive model should map samples further from the training data as they become more out-of-distribution. This can be achieved at two levels: feature level and output level. At the feature level, both methods require the feature extractors (CNNs) to be regularized to prevent feature collapse, which is the mapping of two different data points to the same embedded vector. This is ensured by having input distance awareness, which is equivalent to ensuring bi-Lipschitz continuity over layers of the model [15].

In order to achieve this, DUQ [8] uses a two-sided gradient penalty [16] and SNGP [9] uses bounded spectral normalization [15]. The output level needs to reflect the changes in feature space. This can be done by adopting distance-aware classifiers. DUQ [8] uses a RBF networks with learned centroids for each class and SNGP [9] uses an approximate Gaussian Process layer. We were inspired by temperature scaling [4], which is another method for bettering calibration, but fails under distribution shift [5]. Our method does not require input distance awareness and instead leverages the geometric intuitions about the output layer, specifically properties of the norm of the input embedding, in order to strengthen calibration.

3 Method

Following our hypothesise that the insensitivity of the norm is responsible for bad calibration on distribution shifted data, we propose geometric sensitivity decomposition (GSD) for the norm. We first introduce the geometric perspective of the last linear layer in Sec. 3.1 and then derive GSD in Sec. 3.2. To improve sensitivity of the norm and model calibration on shifted data, we propose a GSD-inspired training and inference procedure in Sec. 3.3 and Sec. 3.4.

3.1 Norm and Similarity

The output layer of a neural network can be written as a dot-product $< \mathbf{x}, \mathbf{w_y} >$, where $\mathbf{x}$ is the embedded input and $\mathbf{w_y}$ is the weight vector associated with class y. Though seemingly simple there are strong geometric and calibration related intuitions drawn from this. Several prior works [10, 12, 11] have studied the effects decomposition of the last linear layer in a softmax model can have on classification. The output layer can be decomposed into angular similarity $\cos\phi_y$ and norm $\|\mathbf{x}\|_2$.

$$P(y|x) = \frac{\exp l_y}{\sum_{j=1}^{c} \exp l_j} = \frac{\exp\left(\|\mathbf{w_y}\|_2 \|\mathbf{x}\|_2 \cos\phi_y\right)}{\sum_{j=1}^{c} \exp\left(\|\mathbf{w_j}\|_2 \|\mathbf{x}\|_2 \cos\phi_j\right)} \tag{1}$$

where $\|\mathbf{w_y}\|_2$ is the norm of a specific classifier in the linear layer. We'll use this geometric view of the linear layer instead of the dot-product representation.

Based on this perspective, we base the foundation of our work on the following observations from prior works [10, 12, 11]: 1) The probability/confidence of the prevalent class of an input is proportional to its norm [12]. 2) While the norm of a feature strongly scales the predictive probability, due to it's unregularized nature the norm is not sensitive to the hardness of the input [11]. In other words, the norm could be the reason for bad sensitivity of the confidence to input distribution shift. Consequently, the insensitive norm can be causally related to bad calibration. We will examine a strong correlation between the quality of calibration and the magnitude of norm in Sec. 4.2.

3.2 Geometric Sensitivity Decomposition of Norm and Angular Similarity

To motivate the subsequent geometric decomposition, we can revisit the softmax model, $P(y|x) \propto \exp\left(\|\mathbf{w_y}\|_2 \|\mathbf{x}\|_2 \cos\phi_y\right)$. There are three terms contributing to the magnitude of the exponential function, $\|\mathbf{w_y}\|_2$, $\|\mathbf{x}\|_2$ and $\cos\phi_y$. Due to weight regularizations, $\|\mathbf{w_y}\|_2$ is most likely very small, while $\cos\phi_y \in [-1, 1]$. Therefore, the only way to obtain a high probability/confidence on training data and minimize cross-entropy loss is to 1) push the norm $\|\mathbf{x}\|_2$ to a large value and 2) keep $\cos|\phi_y|$ of the ground truth class close to one, i.e., $|\phi_y|$ close to zero. This is further supported by [17], where it was shown that logits of the ground truth class must diverge to infinity in order to minimize cross-entropy loss under gradient descent. In this process, models tend towards *large* norms and *small* angles for all training samples.

Therefore, we propose to *decompose* the norms of features into two components: an *instance-independent* scalar offset and an *instance-dependent* variance factor, which we define in Eq. 2. The role of the instance-independent offset $\mathcal{C}_x$ is to minimize the loss on the entire training set and the instance-dependent component Δx accounts for differences in samples. Therefore, if we can disentangle the instance-independent component from the instance-dependent component, we can obtain a norm that is sensitive to the hardness of data. Following this logic, we decompose the norm into two components.

$$\|\mathbf{x}\|_2 = \|\Delta x\|_2 + \mathcal{C}_x \tag{2}$$

Similarly, we relax the angles such that the predicted angular similarity does not need to be close to one on the training data, i.e., making the angles larger. To achieve this, we introduce an instance-independent relaxation angle $\mathcal{C}_\phi$ and an instance-dependent angle $\Delta\phi_y$. Analogous to the norm decomposition, the scalar $\mathcal{C}_\phi$ serves solely to minimize the training loss while the instance-dependent $\Delta\phi_y$ accounts for differences in samples. Because we need to account for the sign of the angle, we put an absolute value on it.

$$|\phi_y| = |\Delta\phi_y| - |\mathcal{C}_\phi| \tag{3}$$

The $\|\Delta\mathbf{x}\|_2$, $|\Delta\phi_y|$ are the instance-dependent components and $\mathcal{C}_x$, $|\mathcal{C}_\phi|$ are the instance-independent components. We can rewrite the pre-softmax logits in Eq. 1 with the decomposed norm and angular similarity. (Detailed derivation in Sec. A.1 in the Appendix.)

$$\begin{aligned}\|\mathbf{x}\|_2 \cos\phi_y = \|\mathbf{x}\|_2 \cos|\phi_y| &= (\|\Delta\mathbf{x}\|_2 + \mathcal{C}_x)\cos\left(|\Delta\phi_y| - |\mathcal{C}_\phi|\right) \\ &= (\|\Delta\mathbf{x}\|_2 + \mathcal{C}_x)\frac{1}{\cos|\mathcal{C}_\phi|}\cos|\Delta\phi_y|\left(1 - \sin|\mathcal{C}_\phi|^2\left(1 - \frac{\cos|\mathcal{C}_\phi|\sin|\Delta\phi_y|}{\sin|\mathcal{C}_\phi|\cos|\Delta\phi_y|}\right)\right)\end{aligned} \tag{4}$$

We can simplify the equation by **assuming** $\cos|\phi_y|$ **is close to one, which means** $|\phi_y|$ **is small.** This is due to the fact that $|\phi_y|$ is the angle between the correct class weight and x, which means as training ensues, the angle converges to 0 and thus the cosine similarity converges to 1. (Please see Sec. A.2 for empirical support.)

$$\frac{\cos|\mathcal{C}_\phi|\sin|\Delta\phi_y|}{\sin|\mathcal{C}_\phi|\cos|\Delta\phi_y|} = \frac{\sin\left(|\Delta\phi_y| + |\mathcal{C}_\phi|\right) + \sin|\phi_y|}{\sin\left(|\Delta\phi_y| + |\mathcal{C}_\phi|\right) - \sin|\phi_y|} \approx 1 \tag{5}$$

Therefore, Eq. 4, omitting the absolute value on angles because *cos* is an even function, simplifies:

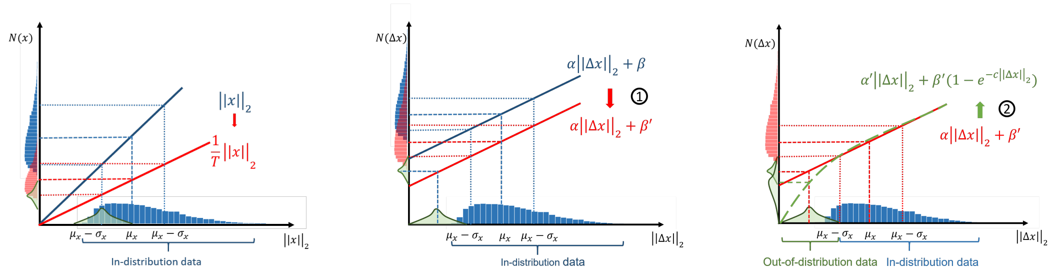
$$\begin{aligned}\|\mathbf{x}\|_2 \cos\phi_y &\approx (\|\Delta\mathbf{x}\|_2 + \mathcal{C}_x)\frac{1}{\cos\mathcal{C}_\phi}\cos\Delta\phi_y \\ &= \left(\frac{1}{\cos\mathcal{C}_\phi}\|\Delta\mathbf{x}\|_2 + \frac{1}{\cos\mathcal{C}_\phi}\mathcal{C}_x\right)\cos\Delta\phi_y \\ &= \left(\frac{1}{\alpha}\|\Delta\mathbf{x}\|_2 + \frac{\beta}{\alpha}\right)\cos\Delta\phi_y\end{aligned} \tag{6}$$

Because $\cos\mathcal{C}_\phi$ and $\mathcal{C}_x$ are instance-independent, we denote them as α and β respectively. **This geometric decomposition of norm and cosine similarity inspires us to include α and β as free trainable parameters in a new network and the network can learn to predict the more input-sensitive $\|\Delta\mathbf{x}\|_2$ and $\Delta\phi_y$ instead of the original $\|\mathbf{x}\|_2$ and ϕ_y.** While both the angle and norm can be decomposed we direct the focus to the norm as the angle is *already* calibrated to accuracy [11]. In other words, angles have been shown to be sensitive to input changes in [11].

3.3 Disentangled Training

Following the derivation in Eq 6, we replace the norm, $\|\mathbf{x}\|_2$, in Eq. 1 by $(\alpha\|\Delta\mathbf{x}\|_2 + \beta)$ and ϕ_y by $\Delta\phi_y$. $\|\Delta\mathbf{x}\|_2$ and $\Delta\phi_y$ are now learned outputs from a new network instead as shown in Eq. 6:

$$P(y|x) = \frac{\exp l_y}{\sum_{j=1}^{c}\exp l_j} = \frac{\exp\left(\|\mathbf{w_y}\|_2\left(\alpha\|\Delta\mathbf{x}\|_2 + \beta\right)\cos\Delta\phi_y\right)}{\sum_{j=1}^{c}\exp\left(\|\mathbf{w_j}\|_2\left(\alpha\|\Delta\mathbf{x}\|_2 + \beta\right)\cos\Delta\phi_j\right)} \tag{7}$$



(a) Temperature Scaling

(b) Ours: calibration Step 1

(c) Ours: Calibration Step 2

Figure 1: **Calibration Procedure** (a): Temperature Scaling [4] changes the slope of the effective norm based on in-distribution (IND) data (See A.9 in Appendix)

The new model can be trained using the same training procedures as the vanilla network without additional hyperparameter tuning, changing the architecture or extended training time. Even though the outputs of the new network, $\|\Delta\mathbf{x}\|_2$ and $\Delta\phi_y$, only approximate the original geometric relationships with Eq. 6, the effect of α and β reflects the decomposition in Eq. 3 and Eq. 2.

- β encodes an instance-independent scalar $\mathcal{C}_x$ of the norm. A larger β corresponds to a smaller instance-dependent component $\|\Delta\mathbf{x}\|_2$.
- α encodes the cosine of a relaxation angle $\mathcal{C}_\phi$. A larger $\arccos\alpha$ corresponds to a larger $\mathcal{C}_\phi$ and therefore a larger $\Delta\phi_j$.

Because β encodes the independent component, the new feature norm $\|\Delta\mathbf{x}\|_2$ becomes sensitive to input changes and maps OOD data to lower norms than IND data as we can see in Fig. 3a, 3b. We regularize α such that the instance-independent component $\mathcal{C}_\phi$ is small. Specifically, we penalize $\|\alpha - 1\|_2^2$ because $\alpha = 1/\cos\mathcal{C}_\phi$, i.e., if $\alpha \approx 1$, $\mathcal{C}_\phi \approx 0$. We empirically found that a larger relaxation angle $\mathcal{C}_\phi$ deteriorates performance because the angular similarity already correlates well with difficulty of data [11] and we do not need to encourage a large relaxation. Sec. 4.3 will empirically verify this argument.

3.4 Disentangled Inference

The decomposition theory in Sec. 3.2 provides a geometric perspective on the sensitivity of the norm and the angular similarity to input changes and inspires a disentangled model in Sec. 3.3. The new model uses a learnable affine transformation on the norm $\|\Delta\mathbf{x}\|_2$. Let's denote the affine transformed norm as the *effective norm* $\mathcal{N}(\Delta\mathbf{x}) \doteq \alpha\|\Delta\mathbf{x}\|_2 + \beta$. However, the training only *separates* the sensitive components of the norm and angular similarity, the model can still be overconfident due to the existence of insensitive components. Therefore, we can improve calibration by modifying insensitive components, e.g., β in our case. We propose a two-step calibration procedure that combines **in-distribution calibration** (Fig. 1b) and **out-of-distribution detection** (Fig. 1c) based on two observations: 1) overconfident IND data can be easily calibrated on a validation set, similar to temperature scaling [4]. 2) for OOD data, without access to a calibration set for OOD data, the best strategy is to map them far away from the IND data given that the model clearly distinguishes them.

The first step is calibrating the model on IND validation set (note our method does *not* rely on OOD validation data), similar to temperature calibration [4]. However, instead of tuning a temperature parameter as shown in Fig. 1a, we simply tune the offset parameter β on the validation set in one of two ways: 1) grid-search based on minimizing Expected Calibration Error (see Sec. 4) 2) SGD optimization based on Negative Log Likelihood [4]. Because these are post-training procedure, both methods are very efficient. We denote the new parameter as β'. As shown in Fig. 1b, by changing the offset, we decrease the magnitude of the norms after the affine transformation. Formally,

$$\mathcal{N}(\Delta\mathbf{x}) = \alpha\|\Delta\mathbf{x}\|_2 + \beta \rightarrow \mathcal{N}(\Delta\mathbf{x}) = \alpha\|\Delta\mathbf{x}\|_2 + \beta' \tag{8}$$

The second step approximates the calibrated affine mapping in Eq. 8 by a non-linear function which covers a wider range of the effective norm as shown in Eq. 9 and maps OOD data further away from IND data. Intuitively, when a sample is more likely IND, the non-linear function maps it closer to the calibrated transformation. When a sample is OOD, the non-linear function maps it more aggressively to a smaller magnitude, exponentially away from the IND samples.

$$\mathcal{N}(\Delta \mathbf{x}) = \alpha \|\Delta \mathbf{x}\|_2 + \beta'(1 - e^{-c\|\Delta \mathbf{x}\|_2}) \tag{9}$$

where c is a hyperparameter which can be calculated as in Eq. 10. The non-linear function grows exponentially close to the calibrated affine mapping in Eq. 8 dictated by $1 - e^{-c\|\Delta \mathbf{x}\|_2}$ as shown in 1c. Therefore, $e^{-c\|\Delta \mathbf{x}\|_2}$ can be viewed as an *error* term that quantifies how close the non-linear function is to the calibrated affine function in Eq. 8. Let μ_x and σ_x denote the mean and standard deviation of the distribution of the norm of IND sample embedding calculated on the validation set. We use the heuristic that when evaluated at one standard deviation below the mean, $\|\Delta \mathbf{x}\|_2 = \mu_x - \sigma_x$, the approximation error $e^{-c(\mu_x - \sigma_x)} = 0.1$. Even though the error threshold is a hyperparameter, using an error of 0.1 lead to state-of-the-art results across all models applied.

$$c = \frac{-ln(1 - error)}{\mu_x - \sigma_x} = \frac{-ln(0.9)}{\mu_x - \sigma_x} \tag{10}$$

In summary, the sensitive norm $\|\Delta \mathbf{x}\|_2$ is used both as a soft threshold for OOD detection and as a criterion for calibration. While similar post-processing calibration procedure exists, such as temperature scaling [4] (illustrated in Fig. 1a and further introduced in A.9) it only provides good calibration on IND data and does not provide any mechanism to improve calibration on shifted data [5]. Our calibration procedure can improve calibration on both IND and OOD data, without access to OOD data, because the training method extracts the sensitive component in a principled manner. Just as temperature scaling, the non-linear mapping needs only to be calculated *once* and adds no computation at inference.

4 Experiments

4.1 Experiments on Calibration

Table 1: **ResNet-28-10 on CIFAR10** averaged over 10 seed. † denotes results from [9]. Our method outperforms other single-pass methods and is comparable to Deep Ensemble [14] on corrupted data. While the ensembled version of our model beats all multi-pass models.

	Method	Accuracy ↑		ECE ↓		NLL ↓	
		Clean	Corrupted	Clean	Corrupted	Clean	Corrupted
Single-Pass	Vanilla†	**96.0±0.01**	72.9±0.01	0.023±0.002	0.153±0.011	0.158±0.01	1.059±0.02
	DUQ†	94.7±0.02	71.6±0.02	0.034±0.002	0.183±0.011	0.239±0.02	1.348±0.01
	SNGP†	95.9±0.01	74.6±0.01	0.018±0.001	0.090±0.012	**0.138±0.01**	0.935±0.01
	Ours β' Grid-Searched	95.9±0.01	**74.9±0.05**	0.018±0.003	**0.067±0.010**	0.148±0.003	**0.826±0.03**
	Ours β' Optimized	95.9±0.01	**74.9±0.05**	**0.008±0.002**	0.085±0.012	0.140±0.004	0.853±0.04
Multi-Pass	Deep Ensembles†	96.6±0.01	**77.9±0.01**	0.010±0.001	0.087±0.004	0.114±0.01	0.815±0.01
	MC Dropout†	96.0±0.01	70.0±0.02	0.021±0.002	0.116±0.009	0.173±0.001	1.152±0.01
	Ours β'Grid-Searched	**96.62**	**77.9**	**0.007**	**0.069**	**0.108**	**0.773**

Table 2: **ResNet-28-10 on CIFAR100** averaged over 10 seeds. † denotes results from [9]. Our method outperforms other single-pass methods and Deep Ensemble [14] on corrupted data. While the ensembled version of our model beats all multi-pass models

	Method†	Accuracy↑		ECE ↓		NLL ↓	
		Clean	Corrupted	Clean	Corrupted	Clean	Corrupted
Single-Pass	Vanilla†	79.8±0.02	**50.5±0.04**	0.085±0.004	0.239±0.020	0.872±0.01	2.756±0.03
	DUQ†	78.5±0.02	50.4±0.02	0.119±0.001	0.281±0.012	0.980±0.02	2.841±0.01
	SNGP†	**79.9±0.03**	49.0±0.02	**0.025±0.012**	0.117±0.014	0.847±0.01	2.626±0.01
	Ours β' Grid-Searched	79.8±0.03	49.8 ± 0.003	0.027±0.003	**0.081 ± 0.007**	0.787±0.009	**2.23±0.02**
	Ours β' Optimized	79.8±0.03	49.8±0.03	0.027±0.003	0.088±0.007	**0.784±0.011**	2.236±0.021
Multi-Pass	Deep Ensembles†	80.2±0.01	**54.1±0.04**	0.021±0.004	0.138±0.013	0.666±0.02	2.281±0.03
	MC Dropout†	79.6±0.02	42.6±0.08	0.050±0.003	0.202±0.010	0.825±0.01	2.881±0.01
	Ours β' Grid-Searched	**83.09**	**54.1**	**0.018**	**0.086**	**0.614**	**2.042**

The ultimate goal of the paper is to improve model calibration under distribution shift by improving sensitivity. Popular metrics for measuring calibration include: Negative Log-Likelihood (**NLL** [18]),

Table 3: **Generalizability Experiments** Our method is effective with different feature backbones.

		Clean				Corrupt/Rotate			
model	dataset	accuracy↑	ECE↓	NLL↓	Brier↓	accuracy↑	ECE↓	NLL↓	Brier↓
ResNet34	CIFAR10	95.63%	0.026	0.186	0.007	**81.96%**	0.164	1.114	0.039
GSD ResNet34	CIFAR10	**95.9%**	**0.005**	**0.148**	**0.006**	76.54%	**0.088**	**0.882**	**0.037**
ResNet50	CIFAR10	95.32%	0.03	0.203	0.008	**76.32%**	0.17	1.23	0.039
GSD ResNet50	CIFAR10	**95.82%**	**0.008**	**0.147**	**0.007**	76.23%	**0.057**	**0.766**	**0.033**
ResNet101	CIFAR10	95.61%	0.028	0.197	**0.007**	77.59%	0.154	1.118	0.037
GSD ResNet101	CIFAR10	**95.62%**	**0.007**	**0.158**	**0.007**	**77.21%**	**0.075**	**0.852**	**0.036**
ResNet152	CIFAR10	**95.7%**	0.028	0.196	**0.007**	75.2%	0.179	1.337	0.041
GSD ResNet152	CIFAR10	95.63%	**0.007**	**0.151**	**0.007**	**76.58%**	**0.058**	**0.765**	**0.033**
ResNet34	CIFAR100	**78.81%**	0.071	**0.868**	**0.003**	**51.16%**	0.19	2.387	**0.007**
GSD ResNet34	CIFAR100	78.02%	**0.037**	0.938	**0.003**	49.27%	**0.098**	**2.361**	**0.007**
ResNet50	CIFAR100	**79.28%**	0.075	**0.861**	**0.003**	49.71%	0.213	2.477	0.007
GSD ResNet50	CIFAR100	78.97%	**0.033**	0.879	**0.003**	**50.12%**	**0.08**	**2.264**	**0.006**
ResNet101	CIFAR100	**80.17%**	0.092	0.846	0.003	**58.19%**	0.253	2.575	0.007
GSD ResNet101	CIFAR100	79.82%	**0.034**	**0.834**	**0.003**	53.14%	**0.082**	**2.11**	**0.006**
ResNet152	CIFAR100	**80.71%**	0.090	**0.815**	**0.003**	**54.2%**	0.233	2.45	0.007
GSD ResNet152	CIFAR100	79.85%	**0.036**	0.827	**0.003**	53%	**0.078**	**2.12**	**0.006**

Table 4: **Importance of Norm** While norm is poorly calibrated, it is important for calibration.

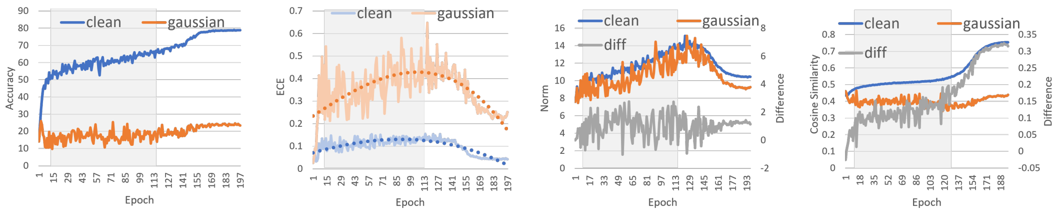
	ECE	NLL	Brier	Entropy	Accuracy
Vanilla ($\|\mathbf{w}_y\|\|\mathbf{x}\| \cos\phi_y$)	0.025±0.001	0.186±0.006	0.001±0.0	0.082±0.002	95.4±0.1%
No Weight Norm (w/o $\|\mathbf{w}_y\|$)	0.061±0.0003	0.206±0.006	0.001±0.0	0.527±0.014	95.4±0.1%
No x Norm (w/o $\|\mathbf{x}\|$)	0.893±0.002	2.837±0.005	0.009±0.0	4.537±0.001	95.4±0.1%
Only Cosine (w/o $\|\mathbf{w}_y\|$,$\|\mathbf{x}\|$)	0.914±0.001	3.235±0.001	0.009±0.0	4.546±0.000	95.3±0.1%

Brier [19] and Expected Calibration Error (**ECE** [20]). Our goal is for our model is to produce values close to 0 in these metrics, which maximizes calibration. Please refer to Sec. A.3 (Appendix) for more detailed discussion on these metrics. Following prior works [9, 8, 5], we will use CIFAR10 and CIFAR100 as the in-distribution training and testing dataset, and apply the image corruption library provided by [1] to benchmark calibration performance under distribution shift. The library provides 16 types of noises with 5 severity scales. In this section, we show that our model outperforms other deterministic methods (despite their significant complexity

Compared Methods We compare to several popular state-of-the-art models including stochastic Bayesian methods (multi-pass): Deep Ensemble [14] and MC dropout [7], and recent deterministic methods (single pass): SNGP [9] and DUQ [8].

Results In Tab. 1 and 2, we compare our model to the most recent state of art deterministic methods SNGP and DUQ using Wide ResNet 28-10 [21] as the model backbone and each model evaluated using the average of 10 seeds. We report accuracy, ECE and NLL on clean and corrupted CIFAR10/100 datasets [1]. Our method outperforms all single-pass methods on calibration when data is corrupted, and even surpass ensembles on error metrics for corrupted data. We had 2 versions of our model: **Grid Searched:** grid search β' on the validation set to minimize ECE and **Optimized:** optimize β' on the validation set via gradient decent to minimize NLL for 10 epochs, similar to temperature scaling. We report additional results with ResNet18 in Sec. A.4 and Sec. A.5 (Appendix) with image noise and rotation respectively.

Generalizability We explored how generalizable our method (Grid Searched) is by applying it to 12 different models and 4 different datasets in Tab. 3. We can see consistently that our model had stronger calibration across all models and metrics, including models known to be well calibrated like LeNet [22]. All models were tested on CIFAR10C and CIFAR100C datasets offered by [1] where the original CIFAR10 and CIFAR100 were pre-corrupted; these were used for consistent corruption benchmarking across all models. All non-CIFAR datasets were corrupted via rotation from angles [0,350] with 10 step angles in between and the average calibration and accuracy was taken across all



(a) CIFAR100 Accuracy (b) CIFAR100 ECE (c) CIFAR100 Norm (d) CIFAR100 Cosine

Figure 2: **Accuracy, ECE, norm and cosine similarity on CIFAR100 validation set with clean and Gaussian noise trained on vanilla ResNet.** In the shaded region, increase in norm is responsible for increase in ECE because cosine similarity is relatively flat. Throughout training, sensitivity of the cosine similarity improves while that of the norm remains insensitive.

Table 5: **Pearson Correlation of Cosine Similarity and Norm vs. ECE during training on CIFAR100**. Norm is consistently positively correlated with ECE whereas the similarity is either negatively or not correlated with ECE.

	ResNet18			ResNet34			ResNet101			ResNet152		
	shot	Gaussian	Defocus	shot	Gaussian	Defocus	shot	Gaussian	Defocus	shot	Gaussian	Defocus
Cosine Sim	0.09	0.03	0.73	0.09	0.03	0.32	-0.03	-0.04	-0.88	-0.97	0.04	-0.81
Norm	**0.82**	**0.82**	**0.78**	**0.82**	**0.81**	**0.78**	**0.87**	**0.87**	**0.85**	**0.86**	**0.85**	**0.81**

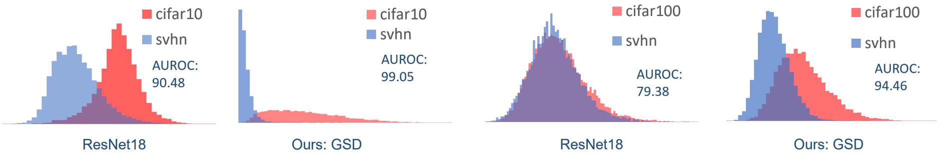
degrees of rotation. Our models included: DenseNet [23], LeNet [22] and 6 varying sizes of ResNet, which are described in [24]. The datasets we experimented on CIFAR10 [25], CIFAR100 [25], MNIST [26] and SVHN [27], CIFAR10C [1], CIFAR100C [1]. We report Optimized results in Tab. 14 in A.7 (Appendix). Both tuning methods yield similar performance.

Qualitative Comparison The current state-of-the-art single pass models for inference on OOD data, without training on OOD data, are SNGP [9] and DUQ [8]. The primary disadvantages of these models are: **1) Hyperparameter Combinatorics:** Both DUQ and SNGP require many hyperparameters as shown in Tab. 13 in A.6 (Appendix). Our model only has *one* hyperparameter that is tuned post-training, which is quicker and less costly than the other methods that require pre-training tuning. **2) Extended Training Time:** DUQ requires a centroid embedding update every epoch, while SNGP requires sampling potentially high dimensional embeddings of training points, thus increasing training time while our model trains in the same amount of time as the model it is applied to. Bayesian MCDO [7] and Deep Ensemble [14] are considered the current state-of-the-art methods for multi-pass calibration. Bayesian MCDO requires multiple passes with dropout during inference. Deep Ensembles requires N times the number of parameters as the single model it is ensembling where N is the number of models ensembled. The main disadvantage of multi-pass models is high inference complexity while our model adds no overhead computation at inference.

Importance of the Norm While we have shown and conjectured that the norm of x is uncalibrated to OOD data and not always well calibrated to IND data, one might suggest to simply remove the norm. We show in Tab. 4 though the norm is uncalibrated it is still important for inference. We trained ResNet18 on CIFAR10 and then ran inference with ResNet18 modified in the following: dividing out the norms of the weights for each class, dividing out the norm of the input and then dividing out both. As we can see the weight norm contributes minimally to inference as accuracy decreased by 0.03% without it and as previous work has shown the angle dominates classification. We can see with $||\mathbf{x}||$ removed the entropy is at it's highest while calibration is very poor, implying the distribution is much more uniform when it should be peaked, as a larger entropy implies a more uniform distribution. Thus the root of the issue does not lie in the existence of the norm, but it's lack of sensitivity.

4.2 Reasons for Bad Calibration under Distribution Shift

To identify the cause of bad calibration, we record the accuracy, ECE, norm and cosine similarity of a model during training of a vanilla ResNet model. Specifically, we record the evaluation statistics on clean data and also on data corrupted with Gaussian noise on CIFAR100. Fig. 2a and 2b show the accuracy and ECE respectively. We observe that evaluation on Gaussian noise corrupted data yields lower accuracy and higher ECE compared to evaluation on clean data. *This demonstrates that*



(a) CIFAR10 vs. SVHN

(b) CIFAR100 vs. SVHN

Figure 3: **Histogram of Norm Distribution** Our model (α-regularized) improves separation of norm between IND and OOD data.

the model's confidence fails to adapt to the decreasing accuracy. Fig. 2c and 2d show the change of average norm and average cosine similarity throughout training. The difference between Gaussian noised data and clean data is also reported. We observe that the norm of clean data and the norm of Gaussian noised data are close and the difference remains constantly low whereas the cosine similarity of the two diverges with training. *This indicates that sensitivity of cosine similarity increases whereas sensitivity of the norm remains low with training.* In the shaded region of Fig. 2b-2d where ECE increases the most, we observe that the norm also increases but the cosine similarity only increases slowly. The observation also holds for other noises and architectures. We further present Pearson correlation between ECE and cosine similarity or norm on 4 models and 3 noises in Tab. 5. A large correlation coefficient indicates a higher positive correlation. Norm is consistently positively correlated with ECE whereas the similarity is either negatively or not correlated with ECE. This shows that the worsening of ECE (large ECE) is correlated with the increasing norm. Based on supporting literature [12], [11] and this correlation, the observation supports the conjecture that the insensitivity of the norm is responsible for bad calibration.

4.3 Empirical Support for the Disentangled Training

Table 6: **OOD AUROC↑ using Norm and Similarity** We show OOD detection results using norm and cosine similarity. SVHN [27] is used as the OOD dataset. Our method (α-regularized) significantly increases the sensitivity of feature norm.

ResNet18	Criterion	CIFAR10	CIFAR10 (Incorrect)
Vanilla	Norm	90.48	67.23
	Similarity	93.87	56.98
α- regularized	Norm	**99.05**	**93.16**
	Similarity	97.09	74.82
α- unregularized	Norm	98.20	88.29
	Similarity	94.72	60.63

(a) **CIFAR10 vs. SVHN AUROC**

ResNet18	Criterion	CIFAR100	CIFAR100 (Incorrect)
vanilla	Norm	79.38	62.66
	Similarity	82.26	55.54
α- regularized	Norm	**94.46**	**86.67**
	Similarity	85.68	63.24
α- unregularized	Norm	84.78	73.11
	Similarity	72.61	42.90

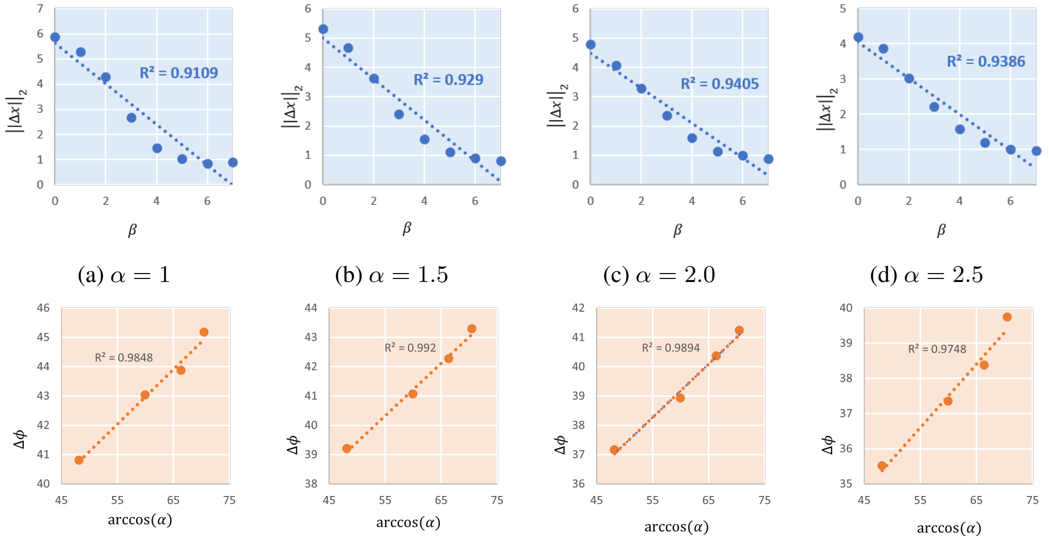
(b) **CIFAR100 vs. SVHN AUROC**

In the first set of experiments, we show that α and β reflect the effects of the geometric decomposition as claimed in Sec. 3.2 with different $\alpha - \beta$ configurations. From Fig. 4a - 4d, we observe that the norm decreases linearly with β for fixed α. From Fig. 4e - 4h, we observe that the angle increases linearly with $arccos(\alpha)$. The observations are consistent with the original geometric motivation. β encodes an instance-independent portion, $\mathcal{C}_x$, of the norm. As β increases, $\mathcal{C}_x$ increases and therefore the magnitude of the dependent component, $\|\Delta x\|_2$ decreases linearly. α encodes the inverse of the cosine of a relaxation angle, $\mathcal{C}_\phi$. As $arccos(\alpha)$ increases, the resulting angle, $\Delta\phi$ increases linearly due to the increased relaxation angle encoded by α.

In the second set of experiments, we show that the new model effectively increases the sensitivity of both the norm and the angle to input distribution shift as claimed in Sec. 3.3. Specifically, we measure OOD detection performance of the models using both the norm and the cosine similarity with the

Table 7: **Average norm and accuracy across different corruptions on GSD ResNet18**. The table is organized in decreasing accuracy order.

ResNet GSD	clean	brightness	fog	elastic	snow	defocus	frost	motion blur	jpeg	zoom blur	pixelate	contrast	shot	glass blur	impulse	Gaussian
accuracy	95.33	93.82	88.75	85.09	83.87	82.95	80.41	79.71	79.31	78.48	76.4	75.29	59.46	59.29	57.26	47.33
norm	0.73	0.66	0.52	0.42	0.46	0.46	0.44	0.37	0.39	0.35	0.5	0.39	0.34	0.27	0.3	0.28



(a) $\alpha = 1$ (b) $\alpha = 1.5$ (c) $\alpha = 2.0$ (d) $\alpha = 2.5$

(e) $\beta = 0$ (f) $\beta = 2$ (g) $\beta = 2$ (h) $\beta = 3$

Figure 4: **Properties of $\|\Delta x\|_2$ and $\Delta\phi$.** (a) - (b): $\|\Delta x\|_2$ decreases linearly with β for fixed α reflecting Eq. 2 and 6. (e) - (h) $\Delta\phi$ increases linearly with $arccos(\alpha)$ for fixed β reflecting Eq. 3 and 6. All plots include R-squared values to indicate goodness-of-fit of the linear relationship.

Area Under the Receiver Operating Characteristic (AUROC) curve metric. We use CIFAR10/100 as the IND data and SVHN [27] as the OOD data. In Tab. 6a and 6b we show two configurations of models in addition to vanilla ResNet18: (α-regularized) we regularize α such that it stays close to one as described in Sec. 3.3; (α-unregularzed) we optimize both α and β freely without constraints. Compared to vanilla ResNet, the norms predicted by our models achieve significant improvement in separating IND data from OOD data. Additionally, we visualize the distribution of norms in Fig. 3a and 3b. The separation between IND and OOD data increases significantly compared to vanilla ResNet18. However, a large α (see α-unregularzed in Tab. 6a and 6b) leads to marginal cosine similarity sensitivity improvement on CIFAR10 and CIFAR100. This indirectly confirms our observations in Sec. 4.2 and in prior works [11] that cosine similarity correlates well with distribution shift. Introducing further angle relaxation might not be always beneficial. While we mainly focus on calibration, our method also strengthens its base model's ability for OOD detection.

The assumption that OOD data have smaller norms is based on the expectation that a model should be less confident on OOD data. Practically, the norm acts as a temperature in softmax as shown in Eq. 1. Intuitively, larger always yields more peaked/confident predictions, and smaller always yields flatter predictive distributions. Therefore, we expect less confident data such as OOD data to have smaller because we expect the output distribution to be flatter. The assumption is supported by the following empirical evidence. In Tab. 7 we show the norm of in-distribution and out-of-distribution data on CIFAR10 using ResNet50-GSD (ours). The OOD data is produced by the 15 corruptions used in the paper. OOD data have consistently smaller norms and the accuracy decreases with decreasing norm with a Pearson correlation of 0.9 as an indicator of more out-of-distribution.

5 Conclusion

In this paper, we studied the geometry of the last linear decision layer and identified the insensitivity of the norm as the culprit of bad calibration under distribution shift. To encourage sensitivity, we derived a general theory to decompose the norm and angular similarity. Inspired by the theory, we proposed a simple yet very effective training and inference scheme that encourages the norm to reflect distribution changes. The model outperforms other deterministic single pass-methods in calibration metrics with much fewer hyperparameters. We also demonstrated its superior generalizability on a variety of popular neural networks. Note that our problem and method have positive societal impact, as calibration under shift improves overall confidence and robustness of these models.

References

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

[2] Sebastian Brechtel, Tobias Gindele, and Rüdiger Dillmann. Probabilistic decision-making under uncertainty for autonomous driving using continuous pomdps. In *17th international IEEE conference on intelligent transportation systems (ITSC)*, pages 392–399. IEEE, 2014.

[3] Yi Yang, Zhigang Ma, Feiping Nie, Xiaojun Chang, and Alexander G Hauptmann. Multi-class active learning by uncertainty sampling with diversity maximization. *International Journal of Computer Vision*, 113(2):113–127, 2015.

[4] 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.

[5] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua V Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *arXiv preprint arXiv:1906.02530*, 2019.

[6] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *arXiv preprint arXiv:1703.04977*, 2017.

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

[8] Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International Conference on Machine Learning*, pages 9690–9700. PMLR, 2020.

[9] Jeremiah Zhe Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax-Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *arXiv preprint arXiv:2006.10108*, 2020.

[10] Weiyang Liu, Zhen Liu, Zhiding Yu, Bo Dai, Rongmei Lin, Yisen Wang, James M Rehg, and Le Song. Decoupled networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2771–2779, 2018.

[11] Beidi Chen, Weiyang Liu, Zhiding Yu, Jan Kautz, Anshumali Shrivastava, Animesh Garg, and Animashree Anandkumar. Angular visual hardness. In *International Conference on Machine Learning*, pages 1637–1648. PMLR, 2020.

[12] Ioannis Kansizoglou, Loukas Bampis, and Antonios Gasteratos. Deep feature space: A geometrical perspective. *arXiv preprint arXiv:2007.00062*, 2020.

[13] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, 2016.

[14] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*, 2016.

[15] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[16] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[17] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. In *The Journal of Machine Learning Research*, page 2822–2878, 2018.

[18] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

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

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

[21] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016.

[22] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[23] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, pages 2261–2269. IEEE Computer Society, 2017.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

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

[26] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[27] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *Advances in Neural Information Processing Systems*, 2011.

[28] Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

[29] Allan H Murphy. A new vector partition of the probability score. *Journal of Applied Meteorology and Climatology*, 12(4):595–600, 1973.

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

A Appendix

A.1 Extended Derivation for Equation 4

In the main paper, we proposed to decompose the norm and angular similarity into instance-independent and dependent components.

$$\begin{aligned} \|\mathbf{x}\|_2 &= \|\Delta x\|_2 + \mathcal{C}_x \\ |\phi_y| &= |\Delta \phi_y| - |\mathcal{C}_\phi| \end{aligned}$$

The $\|\Delta \mathbf{x}\|_2, |\Delta \phi_y|$ are the instance-dependent components and $\mathcal{C}_x, |\mathcal{C}_\phi|$ are the instance-independent components. We can rewrite the pre-softmax logits in Eq. 1 with the decomposed norm and angular similarity.

$$\begin{aligned} \|\mathbf{x}\|_2 \cos \phi_y &= \|\mathbf{x}\|_2 \cos |\phi_y| = (\|\Delta \mathbf{x}\|_2 + \mathcal{C}_x) \cos(|\Delta\phi_y| - |\mathcal{C}_\phi|) \\ &= (\|\Delta \mathbf{x}\|_2 + \mathcal{C}_x)\left(\cos|\Delta\phi_y| \cos|\mathcal{C}_\phi| + \sin|\Delta\phi_y| \sin|\mathcal{C}_\phi|\right) \\ &= (\|\Delta \mathbf{x}\|_2 + \mathcal{C}_x) \frac{1}{\cos|\mathcal{C}_\phi|}\left(\cos|\Delta\phi_y| \cos|\mathcal{C}_\phi|^2 + \sin|\Delta\phi_y| \cos|\mathcal{C}_\phi| \sin|\mathcal{C}_\phi|\right) \\ &= (\|\Delta \mathbf{x}\|_2 + \mathcal{C}_x) \frac{1}{\cos|\mathcal{C}_\phi|} \cos|\Delta\phi_y| \left(\cos|\mathcal{C}_\phi|^2 + \cos|\mathcal{C}_\phi| \sin|\mathcal{C}_\phi| \frac{\sin|\Delta\phi_y|}{\cos|\Delta\phi_y|}\right) \\ &= (\|\Delta \mathbf{x}\|_2 + \mathcal{C}_x) \frac{1}{\cos|\mathcal{C}_\phi|} \cos|\Delta\phi_y| \left(\left(1 - \sin|\mathcal{C}_\phi|^2\right) + \cos|\mathcal{C}_\phi| \sin|\mathcal{C}_\phi| \frac{\sin|\Delta\phi_y|}{\cos|\Delta\phi_y|}\right) \\ &= (\|\Delta \mathbf{x}\|_2 + \mathcal{C}_x) \frac{1}{\cos|\mathcal{C}_\phi|} \cos|\Delta\phi_y| \left(1 - \sin|\mathcal{C}_\phi|^2 \left(1 - \underbrace{\frac{\cos|\mathcal{C}_\phi| \sin|\Delta\phi_y|}{\sin|\mathcal{C}_\phi| \cos|\Delta\phi_y|}}_{\approx 1 \text{ Eq. } 11}\right)\right) \\ &\approx \left(\frac{1}{\cos|\mathcal{C}_\phi|}\|\Delta \mathbf{x}\|_2 + \frac{\mathcal{C}_x}{\cos|\mathcal{C}_\phi|}\right) \cos|\Delta\phi_y| \end{aligned}$$

We can simplify the equation by **assuming** $\cos|\phi_y|$ **is close to one, which means** $|\phi_y|$ **is small.** This is due to the fact that $|\phi_y|$ is the angle between the correct class weight and x, which means as training ensues, the angle converges to 0 and thus the cosine similarity converges to 1. (Please see Sec. A.2 for empirical support.)

$$\frac{\cos|\mathcal{C}_\phi| \sin|\Delta\phi_y|}{\sin|\mathcal{C}_\phi| \cos|\Delta\phi_y|} = \frac{\sin(|\Delta\phi_y| + |\mathcal{C}_\phi|) + \sin|\phi_y|}{\sin(|\Delta\phi_y| + |\mathcal{C}_\phi|) - \sin|\phi_y|} \approx 1 \tag{11}$$

A.2 Small Angle Assumption in Equation 5

Table 8: Average cosine similarity to the ground truth class on the training data set after training for 200 epochs

	CIFAR10			CIFAR100		
	ResNet-18	ResNet-34	ResNet-101	ResNet-18	ResNet-34	ResNet-101
$\cos\phi$	0.81	0.79	0.76	0.75	0.78	0.74

One reason for the small angle assumption in Eq. 5 is the observation that high-capacity models tend to be more miscalibrated [4] and our method is especially more effective in this case. When a model is sufficiently high-capacity compared to the diversity of the dataset, the assumption of small-angle is empirically more valid and the method can provide more significant improvement. All ResNet models are high-capacity deep models and corresponding cosine similarity to the true class is close to one during training as assumed in Sec. 3.2. Tab. 8 shows the average cosine similarity to the ground truth class on the training data.

A.3 Definitions of Metrics

The problem tackled in this paper is supervised image classification in the face of noise. Assume a data point $X_i \in \mathbf{X}, i \in [1, N]$ each associated with a label $Y \in \mathbf{Y} = \{1, ..., K\}$. We would like our model M where $M(X_i) = (\hat{Y}_i, \hat{P}_i)$ where $\hat{Y}_i$ is the class prediction and $\hat{P}$ is the probability/confidence given by the model to be as close to the ground truth distribution $P(Y_i|X_i)$. Ideally $\hat{P}_i$ is well calibrated which means that it represents the likelihood of the true event $\hat{Y}_i = Y_i$. *Perfect calibration* [4] can be defined as:

$$P(\hat{Y}_i = Y_i | \hat{P}_i = P_i) = P_i, \forall P_i \in [0, 1] \tag{12}$$

Ways of evaluating Calibration are as follows:

A.3.1 Expected Calibration Error (ECE)

Expected Calibration Error [20] evaluates calibration by calculating the difference in expectation between the confidence and accuracy or:

$$E_{\hat{P}}[|P(\hat{Y} = Y | \hat{P} = p) - p|] \tag{13}$$

This can also be computed as the weighted average of bins' accuracy/confidence difference:

$$\mathbf{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} |accuracy(B_m) - confidence(B_m)| \tag{14}$$

where n is the total number of samples. Perfect calibration is achieved bins when confidence equals accuracy and ECE $= 0$.

A.3.2 Negative Log Likelihood (NLL)

A way to measure a model's probabilistic quality is to use Negative Log Likelihood [18]. Given a probabilitist model $P(Y|X)$ and N samples it is defined as:

$$\mathbf{L} = -\sum_{i=1}^{N} log(\hat{P}(Y_i|X_i)) \tag{15}$$

where $\hat{P}$ is the predicted distribution of the ground truth P and Y_i is the true label for input X_i. NLL belongs to a class of strictly proper scoring rules [28]. A scoring rule is strictly proper if it is uniquely optimized by only the true distribution. NLL is the negative of the logarithm of the probability of the true outcome. If the true class is assigned a probability of 1, NLL will be minimum with value 0.

A.3.3 Brier

The Brier score [19] measures accuracy of probabilistic predictions. Across all predicted items N in a set of predictions, the Brier score measures the mean squared difference between the predicted probability assigned to possible outcome for $i \in [1, N]$ and the actual outcome.

$$\mathbf{BS} = (1/N) \sum_{t=1}^{N} \sum_{i=1}^{R} (f_{ti} - o_{ti})^2 \tag{16}$$

Where R is number of possible classes, N is overall number of instances of all classes. f_{ti} is the approximated probability of the forecast o_{ti} in one hot encoding. Brier score can be intuitively decomposed into three components: uncertainty, reliability and resolution [29] and it is also a proper scoring rule.

Table 9: **ResNet18 ECE on CIFAR10/100 Noise**, averaged over 5 seeds

	CIFAR10					CIFAR100				
Noise-level	1	2	3	4	5	1	2	3	4	5
Ensemble	0.051	0.075	0.076	0.118	0.184	**0.059**	**0.076**	**0.078**	**0.107**	0.149
MCDO	0.076±0.003	0.098±0.004	0.102±0.002	0.164±0.005	0.251±0.008	0.114±0.002	0.147±0.002	0.14±0.006	0.192±0.009	0.255±0.014
ResNet	0.102±0.001	0.141±0.003	0.153±0.007	0.209±0.011	0.293±0.016	0.113±0.004	0.149±0.005	0.152±0.004	0.185±0.005	0.237±0.01
Ours (GS)	**0.040±0.002**	**0.055±0.003**	**0.060±0.005**	**0.080±0.01**	**0.106±0.012**	0.067±0.002	0.083±0.002	0.089±0.004	0.116±0.007	**0.145±0.013**

Table 10: **ResNet18 NLL on CIFAR10/100 Noise**, averaged over 5 seeds

	CIFAR10					CIFAR100				
Noise-level	1	2	3	4	5	1	2	3	4	5
Ensemble	0.544	0.737	**0.753**	1.055	1.551	**1.499**	**1.927**	**1.969**	**2.379**	**2.99**
MCDO	0.667±0.02	0.845±0.029	0.831±0.013	1.262±0.033	1.947±0.054	1.789±0.011	2.225±0.012	2.236±0.043	2.788±0.072	3.585±0.119
ResNet	0.718±0.01	0.979±0.019	1.043±0.046	1.436±0.081	2.052±0.133	1.782±0.018	2.269±0.023	2.326 ± 0.029	2.773±0.027	3.434±0.032
Ours (GS)	**0.531±0.006**	**0.716±0.012**	0.785±0.013	**1.007±0.018**	**1.346±0.02**	1.786±0.013	2.208±0.013	2.300±0.014	2.676±0.015	3.215±0.028

A.4 Calibration in the Face of Differing Levels of Noise

We report additional calibration ECE, NLL and Brier results in the face of different levels of corruption using ResNet18 in Tab. 9, 10 and 11 respectively. CIFAR10 and CIFAR100's validation set was corrupted using a library of common corruptions [1] with 5 levels of severity. In Tab. 9, 10 and 11 we show how differing levels of common corruptions effect the calibration of models. Across all levels of corruption our model consistently had the stronger Brier score in CIFAR100 and much strong ECE and NLL on CIFAR10.

A.5 Calibration in the Face of Rotation

In Tab. 12b, 12a we rotated CIFAR10 and CIFAR100 validation data set by [0, 350] degrees with 10 degree steps in between, the calibration metrics and accuracy were then averaged. For each model 5 seeds were trained, for MCDO 5 passes were done on each model for inference with a dropout rate of 50% as suggested in the original paper and 5 models were ensembled for Deep Ensemble. β' for our models were 4 on CIFAR10 and 10 for CIFAR100.

A.6 Qualitative Comparison: Extended Discussion

GSD vs. Single Pass Models The current state-of-the-art single pass models for inference on OOD data, without training on OOD data, are SNGP [9] and DUQ [8]. The primary disadvantages of these models is: **1) Hyperparameter Combinatorics:** Both DUQ and SNGP require many hyperparameters as shown in Tab. 13. SNGP requires the most hyperparameters out of all the single pass models. The large combinatoric scale, in addition to the fact that these hyperparameters must be tuned via pre-training grid search, make these methods costly as a full training procedure with multiple epochs are required before evaluating calibration. Our model only has *one* hyperparameter that is tuned post-training with 1 epoch on validation set. **2) Extended Training Time:** DUQ requires a centroid embedding update every epoch, while SNGP requires sampling potentially high dimensional embeddings of training points for generating the covariance matrix as well as updates to the bounded spectral norm on each training step, thus increasing training time while our model trains in the same amount of time as the model it is applied to.

GSD vs. Multi-Pass Models Bayesian MCDO [7] and Deep Ensemble [14] are considered the current state-of-the-art methods for multi-pass calibration. Bayesian MCDO requires multiple passes with dropout during training and inference in order to achieve stronger calibration. Deep Ensembles

Table 11: **ResNet18 Brier on CIFAR10/100 Noise**, averaged over 5 seeds

	CIFAR10					CIFAR100				
Noise-level	1	2	3	4	5	1	2	3	4	5
Ensemble	**0.021**	**0.028**	**0.03**	**0.041**	0.057	0.005	0.006	0.006	0.007	0.008
MCDO	0.024±0.0	0.03±0.001	0.032±0.0	0.046±0.001	0.065±0.001	0.005±0.0	0.006±0.0	0.006±0.0	0.007±0.0	0.009±0.0
ResNet	0.025±0.0	0.034±0.0	0.038±0.001	0.05±0.002	0.068±0.002	0.005±0.0	0.006±0.0	0.007±0.0	0.007±0.0	0.009±0.0
Ours (GS)	0.022±0.0	0.03±0.0	0.034±0.0	0.043±0.001	**0.056±0.001**	**0.003±0.0**	**0.005±0.0**	**0.006±0.0**	**0.007±0.0**	**0.008±0.000**

	ECE↓	NLL↓	Brier↓	Accuracy↑
Ensemble	0.12±0.047	**2.973±0.833**	**0.008±0.002**	**0.338**
MCDO	0.254±0.005	3.78±0.043	0.009±0.0	0.282±0.002
ResNet18	0.215±0.007	3.352±0.036	0.009±0.0	0.311±0.003
Ours	**0.097±0.003**	3.189±0.019	**0.008±0.0**	0.299±0.003

(a) **ResNet18 on CIFAR100 Rotate** over 5 seeds

	ECE↓	NL↓L	Brier↓	Accuracy↑
Ensemble	**0.302**	2.397	**0.082**	**0.44**
MCDO	0.373±0.001	3.08±0.025	0.092±0.0	0.401±0.002
ResNet18	0.42±0.009	2.941±0.085	0.095±0.001	0.427±0.006
Ours	0.323±0.006	**2.211±0.04**	0.085±0.001	0.422±0.005

(b) **ResNet18 on CIFAR10 Rotate** over 5 seeds

Table 13: Model Requirements

Model	Loss Function	Hyperparameters	Output	Multi-pass Infer
Ensemble	CE	Number of Models	LL	True
MCDO	CE	Dropout %	LL	True
SNGP	CE	Spectral Norm Bound, GP scale & bias & discount factor & covariance factor & field factor & ridge penalty	GP	False
DUQ	Multi-BCE	Gradient penalty, RBF sigma, embedding gamma	RBF	False
Ours	CE	β', error (default = 0.1)	Decomposed LL	False

LL: Linear Layer. **CE**: Cross-Entropy, **BCE**: Binary Cross-Entropy, **GP**: Gaussian Process, **RBF**: Radial Basis Function

requires N times the number of parameters as the single model it is ensembling where N is the number of models ensembled. The obvious disadvantage to Deep Ensembles is that it requires N times as long to train and run inference as its base model. While no model currently beats Deep Ensemble in accuracy on both clean data and corrupted data, we have shown that our model has stronger calibration in the face of certain levels of severity of corruption Tab. 1 and 2. Bayesian MCDO has shown to have stronger calibration than the same model not trained with dropout, but tends to suffer large accuracy drops as well as not being as strong as other single pass models or Deep Ensemble in calibration, even with many passes. Our model empirically suffers minimal accuracy drops when compared to its backbone and in some conditions led to stronger accuracy on corrupted data (Tab. 1 and 2).

A.7 Generalizability: Extended Table

Generalizability We explored how generalizable our method is by applying it to 12 different models and 4 different datasets in Tab. 14. We report results for both variants of our model: **Grid Searched:** grid search β' on the validation set to minimize ECE and **Optimized:** optimize β' on the validation set via gradient decent to minimize NLL for 10 epochs, similar to temperature scaling. We can see consistently that our model had stronger calibration across all models and metrics, including models known to be well calibrated like LeNet [22]. All models were tested on CIFAR10C and CIFAR100C datasets offered by [1] where the original CIFAR10 and CIFAR100 were pre-corrupted; these were used for consistent corruption benchmarking across all models. All non-CIFAR datasets were corrupted via rotation from angles [0,350] with 10 step angles in between and the average calibration and accuracy was taken across all degrees of rotation. Our models included: DenseNet [23], LeNet [22] and 6 varying sizes of ResNet, which are described in [24]. The datasets we experimented on CIFAR10 [25], CIFAR100 [25], MNIST [26] and SVHN [27], CIFAR10C [1], CIFAR100C [1].

A.8 Training Parameters and Dataset License

We train all our models using stochastic gradient descent for 200 epochs and a batch size of 128 on RTX 2080 GPUs. We use a starting learning rate of 0.1 and a weight decay of $5.0e-4$. For ResNet18 experiments, we use a cosine scheduler for learning rate. For Wide ResNet-20-10 experiments, we use a step scheduler which multiplies the learning rate at epoch 60, 120 and 160 by 0.2.

The CIFAR10/100 datasets [25] are released under MIT license. The CIFAR10/100C datasets [1] are released under Creative Commons Public license.

A.9 Introduction to Temperature Scaling

Temperature scaling is a simple form of Platt scaling [30]. Temperature scaling uses a scalar T to adjust the confidence of the softmax probability in a classification model. Following the notation from

Table 14: **Extended Generalizability Experiments** We benchmark our method against the vanilla models using 12 different backbones and 4 different dOPTasets. **Grid Searched (GS):** β' grid searched on validOPTion ECE, **Optimized (OPT):** β' optimized via SGD on validation NLL.

		Clean				Corrupt/Rotate			
model	dataset	accuracy↑	ECE↓	NLL↓	Brier↓	accuracy↑	ECE↓	NLL↓	Brier↓
LeNet5	Mnist	96.16%	0.01	0.132	0.006	33.95%	0.43	4.533	0.104
GSD LeNet5 GS	Mnist	**96.86%**	**0.005**	**0.103**	**0.005**	**35.73%**	0.42	4.405	0.101
GSD LeNet5 OPT	Mnist	**96.86%**	0.012	0.106	**0.005**	**35.73%**	**0.406**	**4.173**	**0.01**
DenseNet	SVHN	**41.72%**	0.051	1.71	0.072	14.31%	0.301	3.844	0.107
GSD DenseNet GS	SVHN	41.7%	**0.027**	**1.62**	**0.069**	**14.41%**	0.287	**3.134**	0.106
GSD DenseNet OPT	SVHN	41.7%	0.04	**1.62**	**0.069**	**14.41%**	**0.277**	3.25	**0.105**
ResNet34	CIFAR10	95.63%	0.026	0.186	0.007	**81.96%**	0.164	1.114	0.039
GSD ResNet34 GS	CIFAR10	**95.9%**	**0.005**	**0.148**	**0.006**	**76.54%**	0.088	0.882	0.037
GSD ResNet 34 OPT	CIFAR10	**95.9%**	0.011	0.162	0.007	**76.54%**	**0.054**	**0.813**	**0.035**
ResNet50	CIFAR10	95.32%	0.03	0.203	0.008	**76.32%**	0.17	1.23	0.039
GSD ResNet50 GS	CIFAR10	**95.82%**	**0.008**	**0.147**	**0.007**	76.23%	**0.057**	**0.766**	**0.033**
GSD ResNet50 OPT	CIFAR10	**95.82%**	0.01	0.158	**0.007**	76.32%	0.115	0.928	0.038
ResNet101	CIFAR10	**95.61%**	0.028	0.197	**0.007**	77.59%	0.154	1.118	0.037
GSD ResNet101 GS	CIFAR10	95.62%	**0.007**	0.158	**0.007**	**77.21%**	**0.075**	0.852	0.036
GSD ResNet101 OPT	CIFAR10	95.62%	**0.007**	**0.155**	**0.007**	**77.21%**	0.086	**0.788**	**0.033**
ResNet152	CIFAR10	**95.7%**	0.028	0.196	**0.007**	75.2%	0.179	1.337	0.041
GSD ResNet152 GS	CIFAR10	95.63%	**0.007**	**0.151**	**0.007**	**76.58%**	0.058	0.765	0.033
GSD Resnnet152 OPT	CIFAR10	95.63%	0.01	0.154	**0.007**	**76.58%**	**0.043**	**0.756**	**0.032**
ResNet34	CIFAR100	**78.81%**	0.071	0.868	**0.003**	**51.16%**	0.19	2.387	**0.007**
GSD ResNet34 GS	CIFAR100	78.02%	**0.037**	0.938	**0.003**	49.27%	**0.098**	**2.361**	**0.007**
GSD ResNet34 OPT	CIFAR100	78.02%	0.043	**0.93**	**0.003**	49.27%	0.112	2.372	**0.007**
ResNet50	CIFAR100	**79.28%**	0.0746	0.861	**0.003**	49.71%	0.213	2.477	0.007
GSD ResNet50 GS	CIFAR100	78.97%	**0.0326**	0.879	**0.003**	**50.12%**	**0.08**	**2.264**	**0.006**
GSD ResNet50 OPT	CIFAR100	78.97%	0.041	**0.856**	**0.003**	**50.12%**	0.110	2.28	0.007
ResNet101	CIFAR100	**80.17%**	0.092	0.846	0.003	**58.19%**	0.253	2.575	0.007
GSD ResNet101 GS	CIFAR100	**79.82%**	**0.034**	0.834	**0.003**	53.14%	**0.082**	**2.11**	**0.006**
GSD ResNet101 OPT	CIFAR100	**79.82%**	0.038	**0.829**	**0.003**	53.14%	0.092	2.114	**0.006**
ResNet152	CIFAR100	**80.71%**	0.0895	**0.815**	**0.003**	**54.2%**	0.233	2.45	0.007
GSD ResNet152 GS	CIFAR100	79.85%	**0.0364**	0.827	**0.003**	53%	**0.078**	**2.12**	**0.006**
GSD ResNet152 OPT	CIFAR100	79.85%	0.0397	0.821	**0.003**	53%	0.087	**2.12**	**0.006**

the main paper, let l denotes the logits. The temperature scalar is applied to all classes as following:

$$P(y|x) = \frac{\exp \frac{1}{T} l_y}{\sum_{j=1}^{c} \exp \frac{1}{T} l_j} = \frac{\exp\left(\|\mathbf{w_y}\|_2 \frac{1}{T} \|\mathbf{x}\|_2 \cos \phi_y\right)}{\sum_{j=1}^{c} \exp\left(\|\mathbf{w_j}\|_2 \frac{1}{T} \|\mathbf{x}\|_2 \cos \phi_j\right)} \tag{17}$$

As described in Fig. 1a, the temperature effectively changes the slope of $\|\mathbf{x}\|_2$ from 1 to $\frac{1}{T}$. The temperature parameter is optimized by minimizing negative log likelihood on a validation set while freezing all the other model parameters [4]. Temperature scaling calibrates a model's confidence on IND data and does not change accuracy. However, it does not provide any mechanism to improve calibration on shifted distribution and is inferior to other uncertainty estimation methods in terms of calibration [5].