Mitigating Transformer Overconfidence via Lipschitz Regularization

Wenqian Ye[1,4] **Yunsheng Ma**[2,4] **Xu Cao**[3,4] **Kun Tang**[5]

[1]Department of Computer Science, University of Virginia, Charlottesville, VA, USA
[2]College of Engineering, Purdue University, West Lafayette, IN, USA
[3]Department of Computer Science, University of Illinois Urbana-Champaign, Urbana, IL, USA
[4]AI Lab, Shenzhen Children's Hospital, Shenzhen, China
[5]T Lab, Tencent, Beijing, China

Abstract

Though Transformers have achieved promising results in many computer vision tasks, they tend to be over-confident in predictions, as the standard Dot Product Self-Attention (DPSA) can barely preserve distance for the unbounded input domain. In this work, we fill this gap by proposing a novel Lipschitz Regularized Transformer (LRFormer). Specifically, we present a new similarity function with the distance within Banach Space to ensure the Lipschitzness and also regularize the term by a contractive Lipschitz Bound. The proposed method is analyzed with a theoretical guarantee, providing a rigorous basis for its effectiveness and reliability. Extensive experiments conducted on standard vision benchmarks demonstrate that our method outperforms the state-of-the-art single forward pass approaches in prediction, calibration, and uncertainty estimation.

1 INTRODUCTION

Deep learning (DL) has achieved remarkable performance, making it widely employed in various inference and decision-making systems. However, DL models still make mistakes, making trust and safety an increasingly important topic [Amodei et al., 2016, Jiang et al., 2018], especially in critical applications like self-driving cars [Huang and Chen, 2020] and medical diagnosis [Esteva et al., 2017]. One solution to this problem is for models to not only achieve high accuracy but also refrain from making overly confident predictions.

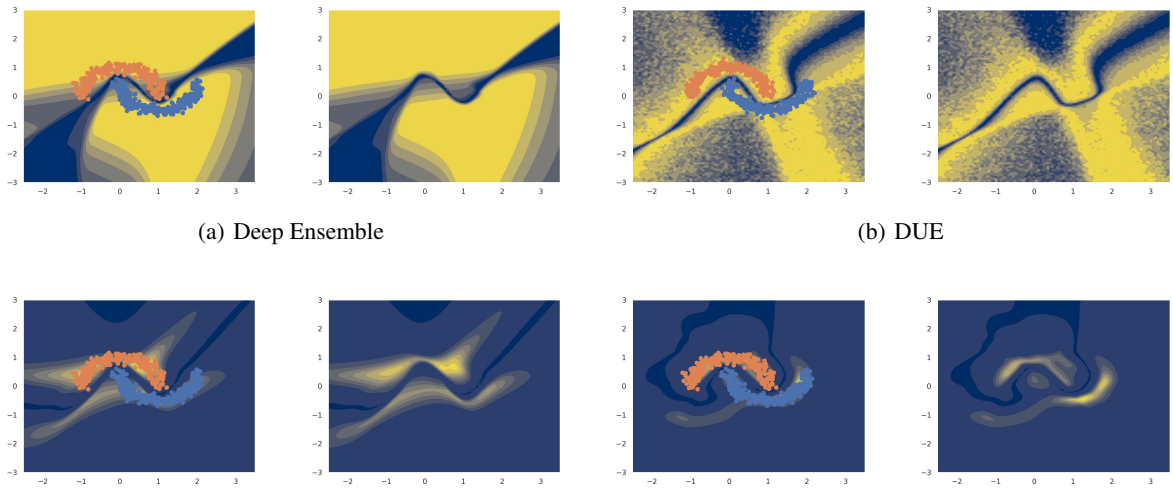
Transformer [Vaswani et al., 2017] and its variants, such as BERT [Devlin et al., 2019], have made significant advances in Natural Language Processing (NLP), especially with large pre-trained models [Gillioz et al., 2020]. Similarly, Vision Transformers (ViT) [Dosovitskiy et al., 2021] and its variants, such as Swin-Transformer [Liu et al., 2021], have recently achieved state-of-the-art performance on a variety of computer vision tasks. In spite of this, their propensity for overconfident predictions is cause for concern, especially as they become the one of the foundation architectures of deep learning. To address this issue, we investigate the under-explored problem of overconfidence issue in Transformers, which can aid subsequent tasks in the construction of reliable models.

Overconfidence is a common problem in many machine learning models for both in- and out-of-distribution inputs, including deep neural networks [Wei et al., 2022, Kristiadi et al., 2020]. When a model is overconfident, it tends to make highly confident predictions even when it is uncertain about the true label of a given input. This can lead to poor performance and inaccurate results, especially in real-world settings where uncertainty is prevalent. Uncertainty estimation is a promising approach for addressing the issue of overconfidence in machine learning models. By estimating uncertainty, a model can make more informed predictions and provide a measure of confidence for each prediction. This can help to improve the robustness and reliability of the model and enable it to perform more effectively in a variety of applications, including decision-making and risk assessment.

Previous techniques to estimating model's predictive uncertainty include Bayesian deep learning [Wilson and Izmailov, 2020, Blundell et al., 2015] and ensemble techniques[Lakshminarayanan et al., 2017, Gal and Ghahramani, 2016]. However, multiple forward passes at the test time are required by most of these methods. In other words, these methods suffer heavy memory and computation cost, which limits their adoption in real-world applications.

Recently, uncertainty quantification via single forward-pass neural networks, which has similar latency as a single deterministic network, has recently received lots of attention [Liu et al., 2020, van Amersfoort et al., 2021, Gillioz et al., 2020]. SNGP [Liu et al., 2020] replaces the dense output

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).



(a) Deep Ensemble (b) DUE

(c) SNGP (d) **LRFormer**

Figure 1: Uncertainty heat map of LRFormer and baseline approaches on the two moons 2D classification benchmark. Orange and blue points are positive and negative training samples respectively. Background color visualizes the predictive uncertainty of each model, where yellow stands for confidence and blue indicates uncertainty. The proposed LRFormer (Figure 1(d)) achieves the closest to ideal uncertainty quantification on this benchmark. Detail refer to Subsection 4.4.

layer with a Gaussian Process (GP) layer and applies Spectral Normalization (SN) [Miyato et al., 2018] to the hidden residual layers. DUE [van Amersfoort et al., 2021] builds upon GPDNN [Bradshaw et al., 2017] and introduces additional constraints to the feature extractor in the form of residual connections in combination with SN [Miyato et al., 2018]. These methods perform well on uncertainty estimation. However, they only focus on bounding the Lipschitz constants of certain CNN modules *i.e.*, convolution and batch normalization [Ioffe and Szegedy, 2015] layers. Moreover, according to Lee et al. [2021], Transformer blocks are very sensitive to the magnitude of Lipschitz constant, and if SN is employed in self-attention modules, training will progress extremely slowly. Although some newest proposed Transformer architectures have been proved to be Lipschitz continuous [Qi et al., 2023, Kim et al., 2021, Dasoulas et al., 2021, Xu et al., 2023, Gupta and Verma, 2023], they still have not solved the overconfidence problem of Transformer.

To address the aforementioned issues, we contribute as follows:

- We propose a novel regularization technique, termed Lipschitz Regularized Self-Attention (LRSA), that addresses distance awareness in both Lipschitzness and Contraction. LRSA replaces the dot product similarity with the distance within Banach Space and normalizes the term by a theoretical bound of the Lipschitz constant. Furthermore, we provide the theoretical analysis of how our method achieves these properties.
- We develop the LRSA based Transformer called LRFormer[1], which integrates distance-preserving hidden mappings in transformer blocks via LRSA and utilizes an optional Gaussian Process (GP) distance-aware output layer for high-quality uncertainty estimation.
- We conduct extensive experiments on widely used OOD benchmarks, including CIFAR-10/-100 versus SVHN and CIFAR-10/-100 versus CIFAR-100/-10. Compared to state-of-the-art approaches, our experimental results demonstrate that the proposed LRFormer model is superior in terms of prediction, calibration, and uncertainty estimation, with minimal time complexity penalty.

2 PROBLEM STATEMENT

In the supervised multi-class classification setting, assume a data sample $(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}$ is sampled from an unknown distribution, where $\mathcal{Y} = \{1, \ldots, K\}$ denote the label space with K classes and $\mathcal{X} = \mathbb{R}^d$ denote the feature space. A learned classifier $f^\theta \colon \mathcal{X} \to \Delta^K$ can produce a probability distribution for $\boldsymbol{x}$ on K classes, where Δ^K is the $K-1$ dimensional unit simplex. In this context, we introduce a former definition of overconfidence for a general classifier.

Definition 2.1 (Overconfidence)**.** Assume f^θ as a composition of a non-probabilistic K-way classifier $\boldsymbol{h}^\theta$ and a softmax function σ, i.e. $\boldsymbol{f}^\theta = \boldsymbol{h}^\theta \circ \sigma$. Given a test data sample $\boldsymbol{x}$, $\boldsymbol{f}^\theta$ provides its probability of assigning it to label i as $\frac{\exp\left(h_i^\theta(\boldsymbol{x})\right)}{\sum_{k=1}^K \exp\left(h_k^\theta(\boldsymbol{x})\right)}$, where $h_i^\theta(\boldsymbol{x})$ denotes the

[1] https://github.com/SZCHAI/LRFormer

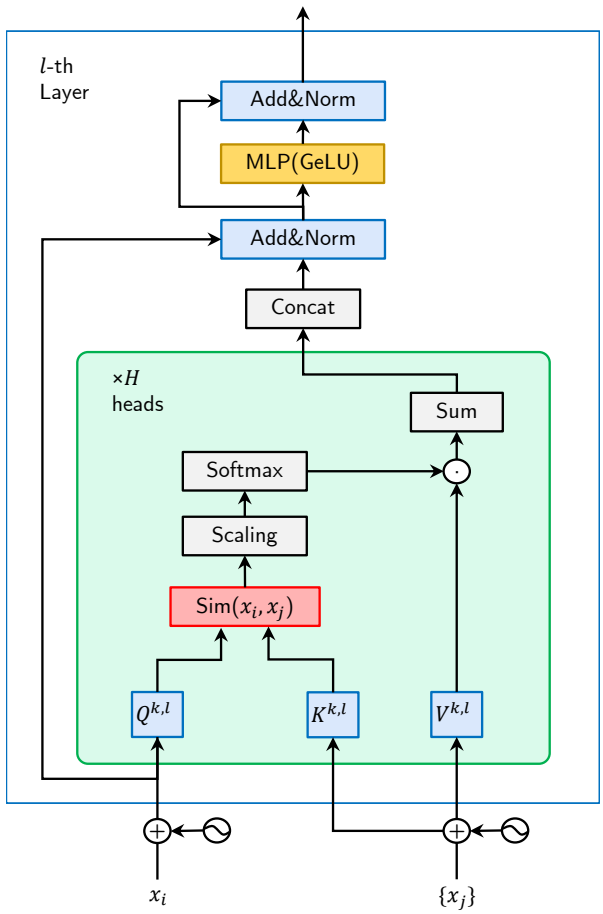


Figure 2: By breaking down a Transformer Layer into its fundamental components, including GeLU Activation, MLP Layer, LayerNorm, and the Attention Module, we gain a comprehensive understanding of their individual contributions to the analysis of Lipschitzness in Transformers. Detailed analysis can be found in Section 3.3.

i-th element of the logit vector produced by $\boldsymbol{h}^\theta$. Then, $\hat{y} := \arg\max_i f_i^\theta(\boldsymbol{x})$ can be returned as the predicted label and $\hat{p} := \max_{j \neq y} f_j^\theta(\boldsymbol{x})$ can be treated as the confidence score. Overconfidence appears when the prediction is wrong with high probability.

Based on Definition 2.1, the logit vector can be decomposed into two components: $\boldsymbol{h}(\boldsymbol{x}) = ||\boldsymbol{h}(\boldsymbol{x})|| \cdot \hat{\boldsymbol{h}(\boldsymbol{x})}$, where $||\boldsymbol{h}(\boldsymbol{x})||$ is the L2-norm of the logit vector and $\hat{\boldsymbol{h}(\boldsymbol{x})}$ is the unit vector in the same direction as $\boldsymbol{h}(\boldsymbol{x})$. These two terms represent the magnitude and direction of the logit vector, respectively. It is evident that if $\arg\max_k(h_k) = c$, then $\arg\max_k(\gamma \boldsymbol{h}_k) = c$ always holds for any given constant value $\gamma > 1$. This indicates that the magnitude of the logit vector does not affect the predicted class c. Additionally, for any given scalar $\gamma > 1$, if $c = \arg\max_k(h_k)$, then $\sigma_c(\gamma \boldsymbol{h}) \geq \sigma_c(\boldsymbol{h})$. From the above claims, we observe that increasing the magnitude $||\boldsymbol{h}(\boldsymbol{x})||$ will lead to a higher softmax confidence score while leaving the final prediction unchanged.

During optimization, the cross-entropy loss is given as:

$$\mathcal{L}_{\text{CE}}(\boldsymbol{h}(\boldsymbol{x};\theta), y) = -\log p(y \mid \boldsymbol{x}) = -\log \frac{e^{||\boldsymbol{h}|| \cdot \hat{h}_y}}{\sum_{i=1}^{k} e^{||\boldsymbol{h}|| \cdot \hat{h}_i}}$$

While the direction $\hat{\boldsymbol{h}(\boldsymbol{x})}$ remains constant, increasing the magnitude will lead to a smaller $p(y \mid \boldsymbol{x})$. In the standard Transformer, optimization on the training loss leads to an increase in the magnitude of the network output to produce a higher softmax confidence score, resulting in a smaller loss.

Lipschitzness can help a classifier tackle the overconfidence issue by limiting the amount of change in the classifier's output when the input is perturbed slightly. When a classifier is overconfident, it tends to assign high confidence to incorrect predictions, which can result in poor performance on unseen data. However, if the classifier is Lipschitz continuous, then the amount of change in its output is limited when the input is perturbed, which can prevent the classifier from making overly confident predictions on Out-of-Distribution (OOD) samples.

Definition 2.2 (Lipschitz Continuity)**.** Lipschitz constant of a function f is an upper bound on the ratio between the output and the input variations of a function f. If $L \in [0, +\infty)$ is such that, for every input $x \in \mathbb{R}^d$ and perturbation $\Delta x \in \mathbb{R}^d$,

$$\|f(x + \Delta x) - f(x)\|_p \leqslant L \|\Delta x\|_p \tag{1}$$

then L is a Lipschitz constant of f. $\|\cdot\|_p$ denotes the p-norm. If X is defined as the ϵ-ball at point x, i.e., $\mathcal{X}' = \{x' \mid \|x - x'\| \leqslant \epsilon\}$, then L is the local Lipschitz constant of f at x.

Furthermore, the Lipschitz condition can be extended to the Bi-Lipschitz condition. Given any two input samples x_1, x_2 and a non-probabilistic K-way classifier $\boldsymbol{h}^\theta$, the Bi-Lipschitz condition can be defined as:

$$L_1 \|x_1 - x_2\| \leq \|\boldsymbol{h}^\theta(x_1) - \boldsymbol{h}^\theta(x_2)\| \leq L_2 \|x_1 - x_2\| \tag{2}$$

where $\|.\|$ is a semantically meaningful distance for the data manifold for positive and bounded Lipschitz constants $0 < L_1 < 1 < L_2$. These bounds L_1, L_2 represent *sensitivity* and *smoothness* conditions which prevents the hidden representations $\boldsymbol{h}^\theta(x)$ from being unnecessarily invariant to the semantically meaningful changes in the input manifold or being overly sensitive to the semantically meaningless perturbations respectively.

Dot-Product Self-Attention is a fundamental building block of the Transformer model. It enables the model to focus on the most relevant parts of the input sequence by weighing the contribution of each input vector to the output, as given by the equation $\mathrm{Attention}(X) = S(X) \cdot V(X) = \mathrm{softmax}\left(\frac{Q \cdot K^\top}{\sqrt{d_k}}\right) \cdot V(X)$. This mechanism can be further generalized by incorporating a similarity function that measures the relevance between input vectors [Katharopoulos et al., 2020]. While several similarity functions, such as the cosine similarity [Qi et al., 2023] or the scaled dot product, have been used in the original formulation, they may not be optimal for all scenarios. Therefore, in this paper, we explore the use of a Lipschitz similarity function to mitigate overconfidence issues in the Transformer. We present a method for constructing a suitable Lipschitz similarity function and demonstrate its effectiveness in improving the robustness and accuracy of the model.

$$\mathrm{Attention}(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{softmax}\left(\frac{\mathrm{sim}(\mathbf{x}_i, \mathbf{x}_j)}{\sqrt{d_k}}\right) \cdot V \quad (3)$$

Our method aims to maintain a reasonable Lipschitz constant at the block level in order to address the issue of overconfidence in neural networks. While other methods, such as Bayesian approaches and label smoothing [Müller et al., 2019], have been proposed to tackle overconfidence, our method incorporates block-wise control. By constraining the Lipschitz constant at each block in the Transformer, we can limit the growth of the magnitude of the network output and reduce overconfidence. This block-wise design also allows our method to be easily integrated into various Transformer-based architectures, including those that have undergone large-scale pretraining. As a result, our method can be scaled up to handle a wide range of tasks and datasets.

3 OUR METHOD

3.1 NOTATIONS AND SETUP

- $S^{(i)} := \mathrm{diag}\left(S_{i:}\right) - S_{i:}^\top, S_{i:} \in \mathbb{R}^{N \times N}$.
- Binary Matrix with one in the (i, j) the entry and zeros elsewhere: $E_{ij} \in \mathbb{R}^{N \times N}$
- Kronecker delta: $\delta_{ij} \in \{0, 1\}$
- $(\infty, 2)$-norm: $\|M\|_{(\infty,2)} = \max_i \left(\sum_j M_{ij}^2\right)^{1/2}$
- Frobenius norm: $\|M\|_F = \left(\sum_{i,j} M_{ij}^2\right)^{1/2}$
- Lipschitz constant $L_{\mathbb{X},\mathbb{Y}}(f)$: for a function $f : \mathbb{X} \to \mathbb{Y}$, $L_{\mathbb{X},\mathbb{Y}}(f) = \sup_{X \in \mathbb{X}} \|\frac{\partial f(X)}{\partial X}\|_{\mathbb{X},\mathbb{Y}}$

3.2 LIPSCHITZ REGULARIZATION ON SELF ATTENTION

Kim et al. [2021] proved that the Scaled Dot-Product Self-Attention does not satisfy the *bi-Lipschitz condition*. To extend the generality of self-attention with high-quality uncertainty estimation, we propose a new regularization method Lipschitz Regularized on Self Attention (LRSA) by replacing the self-attention function with a contractive Bi-Lipschitz expression without losing the original ability of representation. We will explicitly discuss separate aspects to see how to achieve both Lipschitzness and Contraction in our method.

3.2.1 Lipschitzness

Given that *Dot-Product Self-Attention is not Lipschitz*, suppose there exists such mapping $f(X)$, $X \in \mathbb{R}^N$:

$$f(X) = S \cdot X = \mathrm{softmax}(aX \cdot X^\top) \cdot X = \begin{bmatrix} f_1(X) \\ \vdots \\ f_N(X) \end{bmatrix}$$

Its Jacobian Matrix is $J_f = [J_{ij}]_{N \times N}$, each entry can be written as:

$$J_{ij} = aX^\top S^{(i)} \left[E_{ji}X + \delta_{ij}X\right] + S_{ij}I \in \mathbb{R}^{N \times 1}$$

Thus for $i = j$:

$$J_{ii} = aX^\top S^{(i)} E_{ii} X + aX^\top S^{(i)} X + S_{ii} \quad (4)$$

$X^\top S^{(i)} X$ is in the form of a variance of a discrete distribution. When $\mathbf{x}_i = \mathbf{0}$ for some i, some entries of the Jacobian of f grow proportionally to the sample variance of $\mathbf{x}_{\neq i}$.(The softmax probabilities $S_{i:}$ are constant with respect to $\mathbf{x}_{\neq i}$ when $\mathbf{x}_i = 0$.) This will lead to an unbounded Jacobian matrix.

To avoid this pathology, we replace $Q \cdot K^\top$ by $\mathrm{sim}(\mathbf{x}_i, \mathbf{x}_j) = -\|\mathbf{x}_i^\top Q - \mathbf{x}_j^\top K\|_2^2$ in $\mathrm{Attention}(X)$. Here, the new similarity measurement lies on the Banach Space (complete vector space with norm $\|\cdot\|$), which is a more generalized space over Hilbert Space (complete inner product space) [Megginson, 2012]. This modification also gives a strong theoretical guarantee on Lipschitzness with easy matrix multiplications during training.

3.2.2 Contraction

Contraction of the Scaled Dot-Product Self-Attention is another crucial issue for achieving well-calibrated uncertainty. Deriving such contraction scalar requires a theoretical lower bound of Lipschitz constant on the Dot-Product Self-Attention function. A desirable contraction scalar could be non-strict but easy to compute during training.

Theorem 3.1 (Dasoulas et al. [2021]). *For $\alpha \geq 0$, if $\tilde{g}$ is Lipschitz and for all $X \in \mathbb{R}_{d\times n}$, and $\tilde{g}$ satisfy the following conditions:*

1. $\|\tilde{g}(X)\|_\infty \leqslant \alpha c(X)$,
2. $\|X^\top\|_{(\infty,2)} \|\frac{\partial \tilde{g}(X)}{\partial X}\|_{F,(2,\infty)} \leqslant \alpha c(X)$,
3. $\|X^\top\|_{(\infty,2)} \|\frac{\partial c(X)}{\partial X}\|_{F,1} \|\tilde{g}(X)\|_{(2,\infty)} \leqslant \alpha c(X)^2$,

where c is a scalar function $c : \mathbb{R}^{d\times n} \to \mathbb{R}_+$. Then $g(X)$ is Lipschitz:

$$g(X) = \frac{\alpha \tilde{g}(X)}{\max\left\{\|\tilde{g}(X)\|_{(2,\infty)}, \|X^\top\|_{(\infty,2)} L_{F,(2,\infty)}(\tilde{g})\right\}} \tag{5}$$

Inspired by 3.1, we introduce a proper regularization scalar function with a Scalar Factor α by replacing $\tilde{g}(X)$ with $Q \cdot K^\top$:

$$c(X) = \frac{\alpha}{\|Q\|_F \cdot \|X^\top\|_{(\infty,2)}} \tag{6}$$

Here, we assign it as a hyperparameter in control of the corresponding Lipschitz constants for proper contraction of the attention block. Small alpha results in a loss of information while a large alpha causes the model tending to be non-Lipschitz.

3.2.3 Summary

Here is the formal definition of the similarity function:

$$S_{ij} := b \cdot c(X) = -\frac{\alpha \|\mathbf{x}_i^\top W_Q - \mathbf{x}_j^\top W_K\|_2^2}{\|Q\|_F \cdot \|X^\top\|_{(\infty,2)}} \tag{7}$$

This pair-wise operation can alternatively be implemented as a matrix version for improved computational efficiency:

$$S(X) = \text{softmax}(-\alpha \cdot \frac{\|Q\|_{\text{row}}^2 - 2QK^\top + \|K\|_{\text{col}}^{2\top}}{\|Q\|_F \cdot \|X^\top\|_{(\infty,2)}}) \tag{8}$$

LRSA Attention can be represented by the expression $\text{LRSA}(X) = S(X) \cdot V(X)$, where $S(X)$ denotes the similarity scores and $V(X)$ represents the value embeddings. In the following section, we define the Lipschitz Constant of $S(X)$ as L_{LRSA}. From Supplementary Material, we can conclude that L_{LRSA} is bounded by $\frac{6\alpha}{\|X\|_F} \cdot \frac{(\|W_Q\|_2 + \|W_K\|_2)^2}{\|W_Q\|_F}$.

3.3 BOTTOM-UP ANALYSIS ON LRFORMER

Analysis on Lipschitzness of GeLU Activation

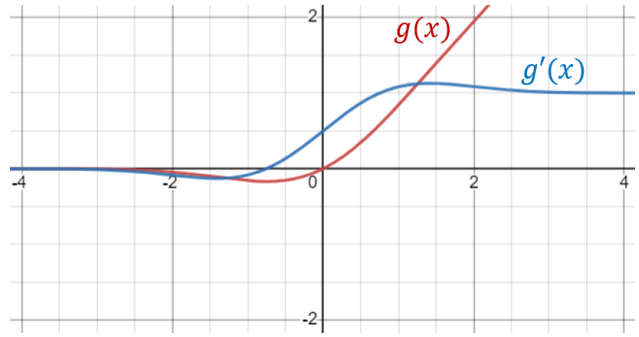


Figure 3: GeLU $g(x)$ and the derivative of GeLU $g'(x)$.

GeLU [Hendrycks and Gimpel, 2016] is the most commonly used activation function in Transformers, especially the GPT series of models and Vision Transformers. GeLU's activation function form is $\text{GeLU}(x) = x\Phi(x)$, where $\Phi(x)$ is the standard Gaussian cumulative distribution function. The derivative of GeLU is given as:

$$\text{GeLU}'(x) = \frac{xe^{-\frac{x^2}{2}}}{\sqrt{2\pi}} + \frac{\text{erf}(\frac{x}{\sqrt{2}})}{2} + \frac{1}{2} \tag{9}$$

,where $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2 dt}$. The second derivative of GeLU is:

$$\text{GeLU}''(x) = \frac{1}{\sqrt{2\pi}}((1+\sqrt{2}) \cdot e^{-\frac{x^2}{2}} - x^2 e^{-\frac{x^2}{2}}) \tag{10}$$

Compared with ReLU, GeLU is differentiable at zero and keeps the Lipschitz continuous. The Lipschitz constant of GeLU is the value of x from $\max(\text{GeLU}'(x))$. Let $\text{GeLU}''(x) = 0$, we can verify $\max(\text{GeLU}'(x)) \approx 1.129$.

Analysis on Lipschitzness of LayerNorm

The raw LayerNorm operation ($\text{LN}(\mathbf{x}) = \frac{\mathbf{x}-\mu(\mathbf{x})}{\sqrt{\sigma^2(\mathbf{x})}} * \boldsymbol{\gamma} + \boldsymbol{\beta}$) [Ba et al., 2016] is not Lipschitz continuous because the ill-defined input with zero variance will lead to a Jacobian matrix filled with infinity.

However, the LayerNorm operation can be changed to a Lipschitz continuous form, which is the LayerNorm used in our models. The form can be expressed as:

$$\text{LayerNorm}(\mathbf{x}) = \frac{\mathbf{x} - \mu(\mathbf{x})}{\sqrt{\sigma^2(\mathbf{x}) + \epsilon}} * \boldsymbol{\gamma} + \boldsymbol{\beta} \tag{11}$$

where $\mathbf{x}, \boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{R}^N$, $\mu(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^N x_i$, $\sigma^2(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^N (x_i - \mu(\mathbf{x}))^2$.

From Supplementary Material, we can conclude that LayerNorm is Lipschitz with the constant $\eta = \epsilon^{-\frac{1}{2}} \max_i |\gamma_i| N$.

Analysis on Lipschitzness of MLP Layer

In each Transformer block, the Attention Module is typically followed by the Multi-Layer Perceptron (MLP) Layer. The MLP layer consists of two Fully Connected (FC) Layers, a Dropout Layer and a GeLU activation function. Since the Dropout Layer has no impact on the Lipschitz constance, it can be simplified as:

$$\mathrm{MLP}(x) = \mathrm{FC}_2 \circ \mathrm{GeLU} \circ \mathrm{FC}_1(x) \tag{12}$$

The upper bound on the Lipschitz constant of a fully-connected layer can be derived by analyzing the effect of each layer independently and considering a product of the resulting spectral norms $\sigma(\cdot)$ [Miyato et al., 2018].

$$L_{\mathrm{MLP}} = 1.129 \cdot \sigma(W_1) \cdot \sigma(W_2) \tag{13}$$

Lipschitz Constant of LRFormer Layer

In the LRFormer, we analyze the LRFormer Layer to demonstrate our method follows the Lipschitzness. Each LRFormer Layer can be expressed as:

$$\begin{aligned}\mathrm{LRFormer}_i(x) = \mathrm{LayerNorm}(\mathrm{LayerNorm}(x + \mathrm{LRSA}(x)) \\ + \mathrm{MLP}(\mathrm{LayerNorm}(x + \mathrm{LRSA}(x))))\end{aligned} \tag{14}$$

Combined with the previous analysis, we conclude the following Lipschitz bound of a single LRFormer layer:

$$\begin{aligned}L_{\mathrm{Layer}} = \eta_1 \cdot \eta_2 \cdot ((1 + L_{\mathrm{LRSA}}) \\ + 1.129 \cdot \sigma(W_1) \cdot \sigma(W_2) \cdot (1 + L_{\mathrm{LRSA}}))\end{aligned} \tag{15}$$

, where η_1, η_2 are the Lipschitz Constant of two LayerNorms.

4 EXPERIMENTS

In this section, we verify the effectiveness of LRFormer in OOD detection with several benchmark datasets. We also design ablation experiments including attention module comparison, searching for a proper scalar factor α and validating the reliability of pretrained models.

4.1 SETUP

4.1.1 Benchmarks

We evaluate the performance of the proposed LRFormer model on the OOD benchmark [Miyato et al., 2018] using SVHN [Netzer et al., 2011] as the OOD dataset for the model trained on CIFAR-10/-100 [Krizhevsky et al., 2009]. OOD data is never seen during training, whereas ID samples are semantically similar to training samples. We also show LRFormer's performance on the Two Moons dataset in Figure 1.

4.1.2 Baselines

Our baselines included the deterministic model and two ensemble models: MC Dropout (with 10 dropout samples) and deep ensembles (with 10 models) [Lakshminarayanan et al., 2017]. All models were trained with a dense output layer and no spectral regularization. Besides, we also compared with three single-model approaches: MCD-GP (with 10 samples), DUQ [Van Amersfoort et al., 2020], DUE [van Amersfoort et al., 2021], and SNGP series [Liu et al., 2020] including DNN-SN and DNN-GP. For models that use a GP layer, we kept DL = 1024 and computed the predictive distribution using Monte Carlo averaging with 10 samples. For fair comparison, we set the backbone with the same parameter magnitude (19.9M parameters for LRFormer, and 36.5M for SNGP series).

4.1.3 Evaluation Metrics

Expected Calibration Error (ECE) [Guo et al., 2017] quantifies the difference between a model's expected confidence (e.g., the maximum probability score) and its actual accuracy. It achieves this by partitioning all the samples, with n representing the total number of samples, into M equally sized bins based on their confidence scores, then calculating the expected difference between accuracy and the average confidence in each bin. In our task, ECE can indicate the effectiveness of the model in dealing with overconfidence.

In addition to ECE, we employ Negative Log Likelihood (NLL), OOD Area Under the Receiver Operating Characteristic Curve (AUROC), and OOD Area Under the Precision-Recall Curve (AUPR) to evaluate the model's performance in overconfidence and uncertainty estimation ability.

4.1.4 Implementation Details

In the following experiments, we resize input image to 224×224 pixels and set the patch size of LRFormer to 16. We employ AdamW [Loshchilov and Hutter, 2017] as the optimizer with a weight decay of 0.05. We use a cosine learning rate scheduler [Loshchilov and Hutter, 2016] with the base learning rate set to 5×10^{-5}. All models are trained for 100 epochs with 10 different random seeds on NVIDIA A100 GPUs.

4.2 COMPARISON WITH STATE-OF-THE-ART MODELS

Following Touvron et al. [2022], we adopt an existing training setup, namely the A3 procedure of Wightman et al. [2021]. We adjust the learning rate of the A3 procedure when training LRFormer. In our experiments, we set the learning rate to 0.006 for LRFormer when pretraining and

Method	Accuracy (↑)	ECE (↓)	NLL (↓)	OOD AUPR (↑)	
				SVHN	CIFAR-100
Deterministic*	96.0 ± 0.01	0.023 ± 0.002	0.158 ± 0.01	0.781 ± 0.01	0.835 ± 0.01
MC Dropout*	96.0 ± 0.01	0.021 ± 0.002	0.173 ± 0.01	0.971 ± 0.01	0.832 ± 0.01
MCD-GP*	95.5 ± 0.02	0.024 ± 0.004	0.172 ± 0.01	0.960 ± 0.01	0.863 ± 0.01
DNN-SN*	96.0 ± 0.01	0.025 ± 0.004	0.171 ± 0.01	0.974 ± 0.01	0.859 ± 0.01
DNN-GP*	95.9 ± 0.02	0.029 ± 0.002	0.221 ± 0.02	0.976 ± 0.01	0.887 ± 0.01
DUQ*	94.7 ± 0.02	0.034 ± 0.002	0.239 ± 0.02	0.973 ± 0.01	0.854 ± 0.01
DUE*	95.6 ± 0.04	0.018 ± 0.002	0.187 ± 0.01	-	-
SNGP*	95.9 ± 0.01	0.018 ± 0.001	0.138 ± 0.01	0.990 ± 0.01	0.905 ± 0.01
Deep Ensemble*†	96.6 ± 0.01	0.010 ± 0.001	0.114 ± 0.01	0.964 ± 0.01	0.888 ± 0.01
LRFormer	$\mathbf{97.2 \pm 0.01}$	$\mathbf{0.012 \pm 0.001}$	$\mathbf{0.100 \pm 0.01}$	$\mathbf{0.993 \pm 0.01}$	$\mathbf{0.911 \pm 0.01}$

Table 1: Comparison between proposed LRFormer and SOTA methods on CIFAR-10 vs SVHN/CIFAR-100 benchmarks, averaged over 10 seeds. The best method among single-network approaches is highlighted in **bold**. *Results from the original papers. † with 10 models.

Method	Accuracy (↑)	ECE (↓)	NLL (↓)	OOD AUPR (↑)	
				SVHN	CIFAR-10
Deterministic*	79.8 ± 0.02	0.085 ± 0.004	0.872 ± 0.01	0.882 ± 0.01	0.745 ± 0.01
MC Dropout*	79.6 ± 0.02	0.050 ± 0.003	0.825 ± 0.01	0.832 ± 0.01	0.757 ± 0.01
MCD-GP*	79.5 ± 0.04	0.085 ± 0.005	0.937 ± 0.01	0.873 ± 0.01	0.754 ± 0.01
DNN-SN*	79.9 ± 0.02	0.098 ± 0.004	0.918 ± 0.01	0.879 ± 0.03	0.745 ± 0.01
DNN-GP*	79.2 ± 0.03	0.064 ± 0.005	0.885 ± 0.01	0.876 ± 0.01	0.746 ± 0.02
DUQ*	78.5 ± 0.03	0.119 ± 0.001	0.980 ± 0.02	0.878 ± 0.01	0.732 ± 0.01
SNGP*	79.9 ± 0.03	0.025 ± 0.012	0.847 ± 0.01	0.923 ± 0.01	$\mathbf{0.801 \pm 0.01}$
Deep Ensemble*†	80.2 ± 0.01	0.021 ± 0.004	0.666 ± 0.02	0.888 ± 0.01	0.780 ± 0.01
LRFormer	$\mathbf{85.2 \pm 0.03}$	$\mathbf{0.018 \pm 0.005}$	$\mathbf{0.538 \pm 0.01}$	$\mathbf{0.955 \pm 0.01}$	0.777 ± 0.01

Table 2: Comparison between proposed LRFormer and the SOTA methods on CIFAR-100 vs SVHN and CIFAR-100 vs CIFAR-10 benchmark, averaged over 10 seeds. The best method among single-network approaches is highlighted in **bold**. *Results from the original papers. † with 10 models.

Method	ECE (↓)	NLL (↓)
DP Attention [Vaswani et al., 2017]	0.066	0.580
L2 Attention [Kim et al., 2021]	0.048	0.582
SCSA [Qi et al., 2023]	0.028	0.626
LRSA	**0.018**	**0.538**

Table 3: Overconfident evaluation comparison among different attention module. The backbone are the same except for the kernel. We compare the checkpoint when the accuracy achieve 0.85 ± 0.01 with CIFAR-100. Note, we select the best hyper-parameter for SCSA ($\nu = 1.0, \tau = 12, \epsilon = 1e-8$)

α	Accuracy (↑)	NLL (↓)
1	0.5637 ± 0.01	1.7063 ± 0.02
100	$\mathbf{0.6601} \pm 0.02$	$\mathbf{1.3712} \pm 0.01$
500	0.6404 ± 0.01	1.4041 ± 0.03
1000	0.6212 ± 0.02	1.4259 ± 0.01

Table 4: Effect of α in LRSA with CIFAR-100

0.004 while finetuning on CIFAR-10/-100. Besides, Different from previous unfair comparison methods [Fort et al., 2021, Xue et al., 2022, Cao and Zhang, 2022], pretrained models from the extra datasets and few-shot outlier exposure settings are not used during training.

To evaluate the model's OOD detection performance, we adopt the two OOD tasks suggested by SNGP: (1) using SVHN as the OOD dataset for a model trained on CIFAR-10/-100; (2) using CIFAR-100/-10 as the OOD dataset for a model trained on CIFAR-10/-100, respectively. Table 1 and Table 2 show the main comparison results. LRFormer outperforms the other single forward pass approaches in all the metrics of CIFAR-10 and most of the metrics of CIFAR-100. Moreover, LRFormer also achieves similar results to Deep Ensemble, which contains 10 models and requires around $10\times$ as much time to execute as LRFormer and other single forward pass approaches.

Dataset	Pretained	Accuracy (↑)	NLL (↓)	OOD AUROC (↑)	OOD AUPR (↑)
CIFAR-10	W/O	0.8528 ± 0.01	0.4447 ± 0.02	0.8500 ± 0.01	0.9078 ± 0.02
	W/	**0.8616 ± 0.01**	**0.4193 ± 0.01**	**0.9125 ± 0.02**	**0.9499 ± 0.01**
CIFAR-100	W/O	0.6404 ± 0.01	1.4041 ± 0.03	0.8421 ± 0.01	0.9165 ± 0.01
	W/	**0.6679 ± 0.01**	**1.2122 ± 0.02**	**0.8689 ± 0.01**	**0.9319 ± 0.01**

Table 5: Ablation study between with (W/) and without (W/O) pre-trained weights from ImageNet-1K dataset. The best method among single-network approaches is highlighted in **bold**. ↓ means lower is better. ↑ means higher is better.

Dataset	Method	Accuracy (↑)	NLL (↓)	OOD AUROC (↑)	OOD AUPR (↑)
CIFAR-10	Transformer	**0.8592 ± 0.01**	0.6972 ± 0.02	0.7552 ± 0.05	0.8521 ± 0.02
	DUE + Transformer	0.8556 ± 0.03	0.5337 ± 0.02	0.8348 ± 0.04	0.8921 ± 0.01
	SNGP + Transformer	0.8542 ± 0.02	0.4933 ± 0.01	0.8275 ± 0.03	0.8960 ± 0.01
	LRFormer	0.8528 ± 0.02	**0.4447 ± 0.01**	**0.8500 ± 0.05**	**0.9078 ± 0.01**
CIFAR-100	Transformer	0.6304 ± 0.02	1.7862 ± 0.01	0.7831 ± 0.02	0.8701 ± 0.02
	SNGP + Transformer	0.6298 ± 0.03	1.5413 ± 0.01	0.8134 ± 0.01	0.8929 ± 0.01
	LRFormer	**0.6404 ± 0.02**	**1.4041 ± 0.02**	**0.8421 ± 0.01**	**0.9165 ± 0.01**

Table 6: Ablation study between the proposed LRFormer and existing uncertainty quantification methods with the same training backbone on the CIFAR-10/-100 vs SVHN benchmark. The best method among single-network approaches is highlighted in **bold**. ↓ means lower is better. ↑ means higher is better.

4.3 ABLATION STUDY

4.3.1 Attention Blocks

To validate the ability to solve overconfidence issues of Transformer, we compare the LRSA with the scale dot-product attention, L2 attention and the scaled cosine similarity attention (SCSA) using overconfidence evaluation metrics. In Table3, we show the best Test ECE and NLL across training for each of the Transformer models. The generalisation performance of the best model for each setting of self-attention is similar.

We find that L2 attention, SCSA, LRSA works well under Lipschitz guarantee. Meanwhile, LRSA works better than the SCSA and L2 attention.

4.3.2 Hyperparameter Analysis

The Scalar Factor α in Equation (6) controls the scale of the Lipschitz constant of the Transformer blocks. In general, we propose running a grid search for $\alpha \in \{..., 100, 500, 1000, ...\}$ to find the highest possible value of α while retaining the predictive performance of LRFormer. In our experiments (Table5), we set scalar factor $\alpha = 1000$ in CIFAR-10, and $\alpha = 500$ in CIFAR-100. The model's performance is not very sensitive to the parameters in the GP output layer, we follow Liu et al. [2020]'s suggestion and set the number of random features to 1024, the length-scale for the RBF kernel to 2, and the L_2 regularization to 0. A proper α value, *i.e.* 100, can preserve both the Lipschitzness and Contraction properties in the model. Small alpha will cause loss of information while large alpha will cause the model tending to be non-Lipschitz, leading to degenerate performance.

4.3.3 Module Comparison

In this section, we compare LRFormer and other OOD detection methods (using Transformer backbone) under the uncertainty estimation setting. We use a depth 6 shallow layer Transformer to conduct this experiments. For Transformer baseline model, we take the predictive entropy as uncertainty. For SNGP + Transformer, the entropy of the average of the Monte Carlo softmax samples is used as uncertainty. We do not compare with DUE for the CIFAR-100 dataset, as its training does not converge. SGD is used as the optimizer with the initial learning rate set to 0.01. All models are trained with batch size 128.

The accuracy, NLL, AUROC, AUPR results are shown in Table 6. The AUROC metric indicates the quality of uncertainty, since it measures the probability that in-distribution (ID) and OOD samples can be separated [Mukhoti et al., 2021]. From the results, we have the following observations:

(1) For OOD detection, *The proposed LRFormer model outperforms all other methods with Transformer backbone on both CIFAR-10 vs SVHN and CIFAR-100 vs SVHN benchmarks*. This superior OOD detection performance benefits from the proposed LRSA regularization method, which solves both Lipschitzness and contraction problems in dot-

product self attention layers, and enables distance preserving mapping in Transformer blocks.

(2) Notably, the superior performance in OOD is achieved without sacrificing LRFormer's predictive performance. On the contrary, LRFormer even outperforms standard Transformer baseline in terms of classifications accuracy on the CIFAR-10 dataset, making LRFormer achieve the best performance in terms of all the metrics compared with all other single-network methods.

(3) Furthermore, the proposed LRSA self-attention can be computed efficiently using matrix operations, with minimal overhead compared to the original dot-product self-attention. This ensures LRFormer's performance gains come without compromising computation cost.

4.3.4 Pre-training

The recent work Plex [Tran et al., 2022] comprehensively validated the reliability of the large pretrained models. The high performance of Transformer results from pre-training on large-size datasets such as ImageNet-21K, and LAION-5B[Schuhmann et al., 2022]. It performs worse than CNNs if trained from scratch on small-size dataset. We use a depth 6 shallow layer Transformer to conduct pretrained weight experiments. The pretrained weights are loaded from standard Transformer, sharing the same weight schemes of MLP layers and position embedding layer. This is because these layers in LRFormer have the same structure as the standard Transformer, so pre-trained weights can be directly applied to them. Our experiments in Table 5 show that LRFormer can also benefit from these pre-trained weights.

In summary, pre-trained weights of models can be directly transferred to LRFormer, which is very convenient for real-world applications.

4.4 VISUALIZATION

In order to show the interpretability of our model, we visualize the uncertainty heat map generated by LRFormer together with the baseline methods on the two moons 2D classification benchmark, which consists of two moon-shaped data clusters separable by a non-linear decision boundary. We employ a tiny Transformer architecture for this task, in which the depth is set to 9, the hidden dimension is set to 24 and the number of heads is set to 8.

The uncertainty heat map comparisons are shown in Figure 1. Background color visualizes the predictive uncertainty of each model, where yellow stands for confidence and blue indicates uncertainty. All methods achieve 100% test accuracy. From the results, we can observe that Deep Ensemble (Figure 1(a)) estimates its uncertainty based on how far away test samples are from the decision boundary, without take the data distribution into consideration. In Figure 1(b), we can see that DUE without restrictions in the feature extractor, produces similar predictive uncertainty to Deep Ensemble which is heavily influenced by the distance from the decision boundary. Although SNGP can make allowances for data distribution, the decision boundary still has an impact on the uncertainty estimation. The proposed LRFormer model, on the other hand, achieves near-ideal uncertainty quantification of this benchmark thanks to its bi-Lipschitz constraint in the LRSA self-attention layers, which allows it to maintain better distance awareness.

5 CONCLUSION

In this paper, we present LRSA, a regularization method designed to address overconfidence issues in Transformer structure models. By enforcing Bi-Lipschitz constraints and self-attention mapping contractions with theoretical guarantees, LRSA encourages the model to generate conservative predictions for out-of-distribution (OOD) inputs, thereby improving its ability to separate in-distribution (ID) data.

Our approach primarily focuses on the attention mechanism of the Transformer architecture, which is a powerful and widely-used component in various natural language processing and vision tasks. Moving forward, it would be beneficial to extend our approach to incorporate these other modules within the Transformer architecture. Exploring how different combinations of modules can be leveraged to enhance performance across various tasks represents a promising avenue for future research. Additionally, investigating the relationship between Lipschitz regularity and other regularization techniques, including weight decay, dropout, and label smoothing, would provide valuable insights. Although these techniques have demonstrated effectiveness in preventing overfitting and improving generalization, their connection to Lipschitz regularity is not yet well-understood. Gaining a deeper understanding of this relationship could unlock insights into the inner workings of deep learning models and potentially lead to further performance improvements.

In conclusion, our proposed LRSA method addresses overconfidence issues in Transformer structure models by encouraging conservative predictions for OOD inputs. While our focus has been on the attention mechanism, future research directions involve incorporating other modules, exploring the relationship between Lipschitz regularity and other regularization techniques, and expanding LRFormer's applicability to diverse models and domains. These efforts contribute to advancing the field of deep learning and improving the robustness and performance of state-of-the-art models.

References

Dario Amodei, Christopher Olah, Jacob Steinhardt, Paul Francis Christiano, John Schulman, and Dandelion Mané. Concrete problems in ai safety. *ArXiv*, abs/1606.06565, 2016.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pages 1613–1622. PMLR, 2015.

J F Bradshaw, Alexander G. de G. Matthews, and Zoubin Ghahramani. Adversarial examples, uncertainty, and transfer testing robustness in gaussian process hybrid deep networks. *arXiv: Machine Learning*, 2017.

Senqi Cao and Zhongfei Zhang. Deep hybrid models for out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4733–4743, 2022.

George Dasoulas, Kevin Scaman, and Aladin Virmaux. Lipschitz normalization for self-attention layers with application to graph neural networks. *arXiv preprint arXiv:2103.04886*, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2021.

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

Stanislav Fort, Jie Ren, and Balaji Lakshminarayanan. Exploring the limits of out-of-distribution detection. *Advances in Neural Information Processing Systems*, 34: 7068–7081, 2021.

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.

Anthony Gillioz, Jacky Casas, Elena Mugellini, and Omar Abou Khaled. Overview of the transformer-based models for nlp tasks. *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, pages 179–183, 2020.

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.

Kavya Gupta and Sagar Verma. Certvit: Certified robustness of pre-trained vision transformers. *arXiv preprint arXiv:2302.10287*, 2023.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv: Learning*, 2016.

Yu Huang and Yue Chen. Autonomous driving with deep learning: A survey of state-of-art technologies. *ArXiv*, abs/2006.06091, 2020.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ArXiv*, abs/1502.03167, 2015.

Heinrich Jiang, Been Kim, and Maya R. Gupta. To trust or not to trust a classifier. In *NeurIPS*, 2018.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020.

Hyunjik Kim, George Papamakarios, and Andriy Mnih. The lipschitz constant of self-attention. In *ICML*, 2021.

Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being bayesian, even just a bit, fixes overconfidence in relu networks. In *International conference on machine learning*, pages 5436–5446. PMLR, 2020.

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

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NIPS*, 2017.

Kwonjoon Lee, Huiwen Chang, Lu Jiang, Han Zhang, Zhuowen Tu, and Ce Liu. Vitgan: Training gans with vision transformers. In *International Conference on Learning Representations*, 2021.

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*, abs/2006.10108, 2020.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Robert E Megginson. *An introduction to Banach space theory*, volume 183. Springer Science & Business Media, 2012.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *ArXiv*, abs/1802.05957, 2018.

Jishnu Mukhoti, Andreas Kirsch, Joost R. van Amersfoort, Philip H. S. Torr, and Yarin Gal. Deterministic neural networks with inductive biases capture epistemic and aleatoric uncertainty. 2021.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

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

Xianbiao Qi, Jianan Wang, Yihao Chen, Yukai Shi, and Lei Zhang. Lipsformer: Introducing lipschitz continuity to vision transformers. In *International Conference on Learning Representations*, 2023.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022.

Hugo Touvron, Matthieu Cord, Alaaeldin El-Nouby, Jakob Verbeek, and Herv'e J'egou. Three things everyone should know about vision transformers. *ArXiv*, abs/2203.09795, 2022.

Dustin Tran, Jeremiah Liu, Michael W Dusenberry, Du Phan, Mark Collier, Jie Ren, Kehang Han, Zi Wang, Zelda Mariet, Huiyi Hu, et al. Plex: Towards reliability using pretrained large model extensions. *arXiv preprint arXiv:2207.07411*, 2022.

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.

Joost R. van Amersfoort, Lewis Smith, Andrew Jesson, Oscar Key, and Yarin Gal. On feature collapse and deep kernel learning for single forward pass uncertainty. 2021.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *ArXiv*, abs/1706.03762, 2017.

Hongxin Wei, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, and Yixuan Li. Mitigating neural network overconfidence with logit normalization. In *International Conference on Machine Learning*, pages 23631–23644. PMLR, 2022.

Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *arXiv e-prints*, pages arXiv–2110, 2021.

Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *ArXiv*, abs/2002.08791, 2020.

Xiaojun Xu, Linyi Li, Yu Cheng, Subhabrata Mukherjee, Ahmed Hassan Awadallah, and Bo Li. Certifiably robust transformers with 1-lipschitz self-attention. 2023.

Feng Xue, Zi He, Chuanlong Xie, Falong Tan, and Zhenguo Li. Boosting out-of-distribution detection with multiple pre-trained models. *arXiv preprint arXiv:2212.12720*, 2022.