Out of Distribution Detection via Neural Network Anchoring

Rushil Anirudh ANIRUDH1@LLNL.GOV
Jayaraman J. Thiagarajan JJAYARAM@LLNL.GOV
Center for Applied Scientific Computing (CASC),
Lawrence Livermore National Laboratory

Editors: Emtiyaz Khan and Mehmet Gönen

Abstract

Our goal in this paper is to exploit heteroscedastic temperature scaling as a calibration strategy for out of distribution (OOD) detection. Heteroscedasticity here refers to the fact that the optimal temperature parameter for each sample can be different, as opposed to conventional approaches that use the same value for the entire distribution. To enable this, we propose a new training strategy called anchoring that can estimate appropriate temperature values for each sample, leading to state-of-the-art OOD detection performance across several benchmarks. Using NTK theory, we show that this temperature function estimate is closely linked to the epistemic uncertainty of the classifier, which explains its behavior. In contrast to some of the best-performing OOD detection approaches, our method does not require exposure to additional outlier datasets, custom calibration objectives, or model ensembling. Through empirical studies with different OOD detection settings – far OOD, near OOD, and semantically coherent OOD - we establish a highly effective OOD detection approach. Code to reproduce our results is available at github.com/LLNL/AMP
Keywords: OOD Detection, Temperature Scaling, Calibration, Anchoring, Uncertainty

1. Introduction

The task of using a trained model to accurately distinguish between samples from the dataset used for training – *i.e.*, the in-distribution (ID), and any other external dataset with different semantic characteristics is broadly referred to as OOD (out-of-distribution) detection. To solve this challenging problem, one needs to obtain an effective characterization of the ID data manifold, such that the discrepancy between test data and the *inferred manifold* can be used to recognize the model's lack of knowledge about OOD data. This is commonly achieved by learning a scoring function: $\mathcal{S} : X \rightarrow \mathbb{R}$ that can score both ID and OOD samples appropriately. A simple scoring function can be based on the maximum softmax probability (MSP) of a prediction, with the expectation that the model will be more confident about an ID sample compared to OOD samples. However, in practice, such simple prediction confidence scores are poorly calibrated, and as a result, several novel scoring functions have emerged – predictive entropy (Guo et al., 2017), energy (Liu et al., 2020), uncertainty estimates (Gal and Ghahramani, 2016; Lakshminarayanan et al., 2017), latent space deviation (Van Amersfoort et al., 2020), class-specific deviations (Lee et al., 2018b; Sastry and Oore, 2020) etc. Though these scoring functions often perform better than MSP, many state-of-the-art formulations (Hendrycks et al., 2019; Liu et al., 2020; Yang et al., 2021) rely on additional unlabeled data for calibrating model predictions to better

reject OOD data. In addition to requiring sophisticated training strategies (e.g., outlier exposure), this approach can be sub-optimal when the calibration dataset is not *strictly* OOD, *i.e.*, and they contain shared semantics with the ID set (Yang et al., 2021). Further, the calibration strategy used in many of these methods relies on *temperature scaling* (Guo et al., 2017), which essentially scales the logits by a scalar called the temperature. When the temperature parameter is greater (or lower) than 1, the entropy of the resulting prediction distribution increases (or decreases). Consequently, with an appropriate temperature value (chosen with either external or additional validation data), even this simple scaling leads to much improved OOD detection performance.

Heteroscedastic temperature scaling with anchoring. In this paper, we explore the idea of *heteroscedastic* temperature scaling, *i.e.*, instead of using the same temperature scalar for all the samples, we construct a temperature function that produces sample-specific temperature values. Our hypothesis is that by appropriately tempering the predictions for ID and OOD samples, any existing scoring function can effectively between distinguish them. We achieve this using a novel training procedure called *neural network anchoring*. In a nutshell, anchoring involves first transforming the input image, x, into a tuple using the transformation $\mathcal{E} : \mathrm{x} \rightarrow [\mathrm{c}, \mathrm{x} - \mathrm{c}]$, where c is another randomly chosen image ("anchor") from the training set, and predicting the label for x using this tuple. We also propose an additional consistency training strategy by perturbing the anchor before encoding, which boosts the performance further. During inference, we obtain predictions from multiple random anchors and propose to estimate the temperature based on standard deviation of these predictions. Using neural tangent kernel theory (Jacot et al., 2018), we show that our heteroscedastic temperature estimate is closely related to the *epistemic* uncertainty of the model.

We use this temperature estimate to calibrate the predictions for a test sample, using which we can compute an OOD score using existing scoring functions (e.g., entropy). See Fig. 1(A) for an illustration of the process, and Figs. 2 and 3 for the pseudo-codes. Fig. 1(B) illustrates the improvement over conventional temperature scaling using the standard CIFAR-10/SVHN OOD benchmark. Through extensive empirical analysis, we demonstrate that the proposed approach produces state-of-the-art detection performance across multiple benchmarks and models (summarized in Table 1).

Base Model	**Benchmark** (IN $\rightarrow$ OOD)	Year	**Reference**
WRN-40-2	CIFAR-10/100 $\rightarrow$ 6 Datasets	(Liang et al., ICLR'17)	Table 2
ResNet-34	CIFAR-10/100 $\rightarrow$ 7 Datasets	(Sastry and Oore, ICML'20)	Table 3
ResNet-34	CIFAR-10 $\leftrightarrow$ CIFAR-100	(Near OOD)	Table 4
ResNet-50	ImageNet-1K $\rightarrow$ ImageNet-C (Krishnan and Tickoo, NeurIPS'20)		Table 5, Figure 5
ResNet-18	Semantically Coherent OOD	(Yang et al., ICCV'21)	Table 6
ResNet-34	Robustness to resizing artifacts	(this paper)	Table 7
WRN-40-2	Ablation study		Table 8

Table 1: Summary of experiments in this paper.

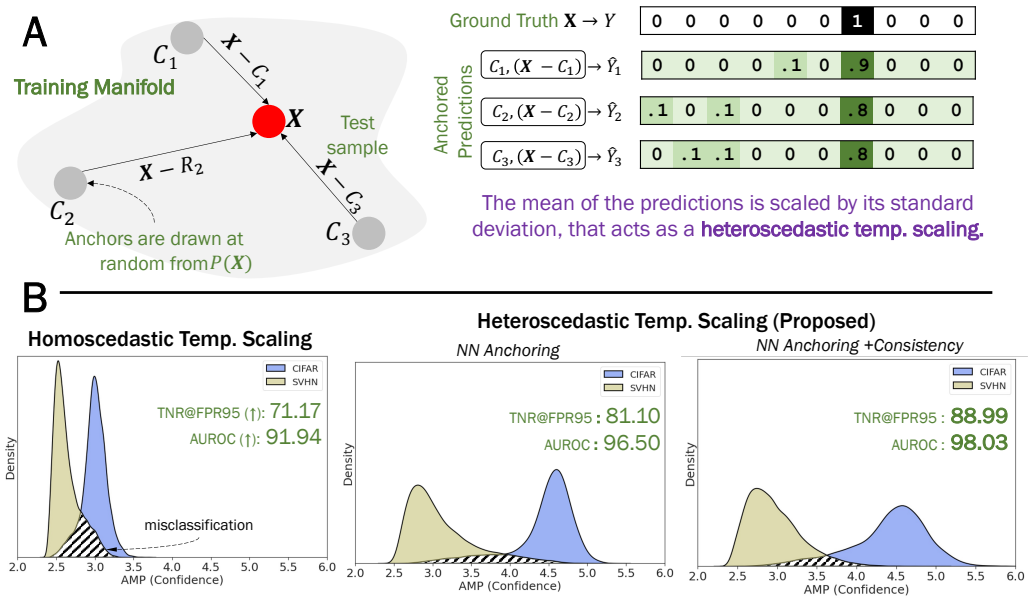


Figure 1: **Improving existing OOD detectors via heteroscedastic temperature scaling.** (A) We propose a new training procedure called neural network anchoring to estimate the temperature parameter for any test sample, and show that it can be leveraged to improve conventional OOD detectors (e.g., entropy-based). (B) We also introduce a new consistency training objective to further improve the fidelity of detectors.

2. Background and Related Work

We use training data $\mathcal{D} = \{(\mathrm{x}_i, y_i)\}_{i=1}^{n}$, where $\mathrm{x}_i \in P_I$ and $y_i \in \mathcal{C}_I := \{1, 2, \cdots, N_{class}\}$, to train a model $f(\boldsymbol{\theta}) \in \mathcal{H}$ with randomly initialized weights $\boldsymbol{\theta}_0$ and hypothesis space $\mathcal{H}$.We train a classifier $f(\theta) : \mathrm{x} \to y$, parameterized by θ using $\mathcal{D}$. While the learned model is required to generalize to the test dataset $\mathcal{D}^t = \{(\mathrm{x}_i^t, y_i^t)\}_{i=1}^{N_t}$ when $\mathrm{x}_i^t \in P_I$ and $y_i^t \in \mathcal{C}_I$, it is also critical to recognize out-of-distribution samples, *i.e.*, $\mathrm{x}_i^t \in P_O$ and $y_i^t \in \mathcal{C}_O$, on which the model could fail. Without loss of generality, this formulation encompasses scenarios with unknown distribution shifts to the input images, *i.e.*, $P_O \neq P_I, \mathcal{C}_O \subseteq \mathcal{C}_I$ as well as the presence of additional, unknown classes $\mathcal{C}_O \supset \mathcal{C}_I$.

Scoring functions In summary, the goal of OOD detection is to use labeled data $\mathcal{D}$ to train a classifier that has the capacity to reject samples from P_O, while also accurately classifying samples from P_I. This is typically achieved by defining a scalar *scoring function*, such that $\mathcal{S}_{\mathrm{x}\in P_I} \neq \mathcal{S}_{\mathrm{x}\in P_O}$, i.e., they are sufficiently distinct that an out of distribution sample is easily classified. Choosing the appropriate scoring function has been focus of the last several years of research, with most techniques choosing the score as some function of the prediction logits obtained from the trained classifier – such as maximum softmax probability, entropy (Guo et al., 2017), and energy (Liu et al., 2020).

Uncertainties for OOD detection Uncertainty estimation has been a popular choice for OOD detection since epistemic (or model) uncertainties are supposed to be indicative of the OOD-ness of a test sample. For example, DUQ (Van Amersfoort et al., 2020) uses a

kernel distance to a set of class-specific centroids defined in the feature of a deep network as the measure for uncertainty. Another recent technique is DEUP (Jain et al., 2021) which trains an explicit epistemic uncertainty estimator for a pre-trained model. More generally, Bayesian methods (Neal, 2012) are among the most common kinds of uncertainty estimators today, but they are not easily scalable to large datasets and are known to be outperformed by model ensembling (Lakshminarayanan et al., 2017). Monte Carlo Dropout (Gal and Ghahramani, 2016) is a scalable alternative to Bayesian methods in that it approximates the posterior distribution on the weights via dropout to estimate uncertainties.

Prediction calibration for OOD detection In general uncertainty-based OOD detectors have so far not been able to improve performance over more traditional scoring based methods. As a result, a lot of focus has been on calibrating classifier predictions to make the scoring function more effective (often relying on outlier examples) such as Mahalanobis (Lee et al., 2018a), ODIN (Liang et al., 2017), Gram matrices (Sastry and Oore, 2020), AVuC (Krishnan and Tickoo, 2020), and outlier exposure (Hendrycks et al., 2019). The simplest, and often most effective, strategy for calibration is temperature scaling (Guo et al., 2017) where the logits are scaled by a temperature parameter (τ). When $\tau > 1$ the prediction is made less confident (i.e., higher entropy), and if not it is made more peaky (i.e., lower entropy). Notably, most techniques define a single τ value to calibrate predictions on an entire test set.

The underlying assumption made by such approaches is that the predictions are *homoscedastic* – i.e., the errors (and uncertainties) are uniform everywhere and therefore a single scalar for all samples suffices. However, in practice, most models are *heteroscedastic* – i.e., the errors around a prediction can vary for different test samples. In this paper, we propose a novel calibration strategy that estimates a specific temperature for every sample related to the unreliability or uncertainty for that sample. Unlike existing approaches, this takes the heteroscedasticity of the model into account and therefore, subsequently improves OOD detection.

3. Heteroscedastic Temperature Scaling via Neural Network Anchoring

In this section, we first outline the proposed scoring function, followed by theoretical and empirical justification of its effectiveness in visual OOD detection. As previously stated, we are interested in learning a temperature function $\boldsymbol{\tau} : \mathcal{X} \rightarrow \mathbb{R}$, defined in the image domain $\mathcal{X}$, that is used to calibrate a model's predictions. The temperature parameter for a sample x is denoted as $\boldsymbol{\tau}(\mathrm{x})$ (fixed to be the same for all classes). We refer to this process as heteroscedastic scaling, since the model predictions for any test sample can be adjusted with the learned function. Next, we discuss the proposed approach for constructing $\boldsymbol{\tau}$.

3.1. Neural network anchoring

First, let us randomly choose a training image from the dataset $\mathcal{D}$, denoted by c and refer to it as an *anchor*. Using this anchor c, we define a simple coordinate transformation on the input domain as $\mathcal{E} : \mathrm{x} \rightarrow [\mathrm{c}, \mathrm{x} - \mathrm{c}]$. That is, we represent an image as a combination of the anchor, and the residual between the anchor and the image. Note that, this definition allows the use of multiple transformations (w.r.t. many anchors) to obtain predictions for

a given sample x, *i.e.*, $f_A([\mathrm{c}_1, \mathrm{x} - \mathrm{c}_1]) = f_A([\mathrm{c}_2, \mathrm{x} - \mathrm{c}_2]) = \cdots = f_A([\mathrm{c}_k, \mathrm{x} - \mathrm{c}_k])$, where f_A refers to the model that takes the tuple $([\mathrm{c}_k, \mathrm{x} - \mathrm{c}_k])$ and predicts the target y.

Training. During training, for every input sample x_i, we use one random anchor to perform the coordinate transformation, which is implemented as a simple concatenation along the channel dimension. Due to the randomness over the choice of the anchor, over the course of training each x_i gets combined with a large number of anchors. Since the prediction for the sample x_i – regardless of the choice c – is expected to be the same (label y_i), this enforces an implicit consistency in the predictions across different anchors. The optimization of this anchored model is similar to that of a standard network, e.g., cross entropy loss-based training.

Consistency via standard image augmentations. Inspired by the recent successes of data augmentation strategies in improving model generalization, we exploit an additional consistency during training to further improve anchored models. More specifically, we modify the input to be as follows – $\bar{\mathrm{x}} = [\mathcal{T}(\mathrm{c}), \mathrm{x} - \mathrm{c}]$, where $\mathcal{T}(.)$ refers to a pre-defined image augmentation and it returns a perturbed anchor. Intuitively, we encourage the model to learn invariances between c and $\mathcal{T}(\mathrm{c})$ using an asymmetry in the coordinate transformation. While different choices currently exist to implement $\mathcal{T}$ for natural images, we find that using a composition of multiple standard augmentation strategies already used in training the model (random crops, random flips, color jitter etc.) are sufficient in practice.

Figure 2: Pseudocode for training

```
def train_loop(trainloader,T):
    for inputs, targets in trainloader:
        A = Shuffle(inputs)
        D = inputs-A
        X_d = torch.cat([T(A), D],axis=1)
        y_d = model(X_d)
        loss = criterion(y_d,targets)
        ....

return
```

Figure 3: Pseudocode for inference

```
def inference(inputs,anchors):
    for A in anchors:
        D = inputs-A
        X_test = torch.cat([A, D],axis=1)
        y_test = model(X_test)
        preds.append(y_test)
    P = torch.cat(preds,0)
    H = torch.mean(P,0)
    tau = P.sigmoid().std(0).sum(1)
    ood_score = AMP(H/tau)
return H, ood_score
```

Inference. As discussed in the section, this anchoring process leads to different hypotheses for different anchor choices, so we propose to marginalize out the effect of the anchor, c, to both obtain the predictions as well as the temperature function for performing the heteroscedastic calibration. The temperature value for a sample is estimated as the standard deviation of the predictions, obtained using multiple random anchor choices, as shown below:

$$H(y|\mathrm{x}) = \mathsf{MEAN}\left[f_A([\mathrm{c}_k, \mathrm{x} - \mathrm{c}_k])\right]_{k=1}^{K}; \boldsymbol{\tau}(\mathrm{x}) = \sum_{\text{all classes}} \mathsf{STD\text{-}DEV}\left[\sigma\left(f_A([\mathrm{c}_k, \mathrm{x} - \mathrm{c}_k])\right)\right]_{k=1}^{K}. \tag{1}$$

Here, for convenience we use MEAN and STD-DEV to denote the mean and standard deviation over predictions obtained using K different anchors during inference. Further, $H(y|\mathrm{x})$ indicates the set of logits for each of the classes. To scale the standard deviation appropriately, we compute it after passing the logits through a sigmoid activation layer. Note that, we compute the class-specific uncertainties by performing this aggregation directly using the logits obtained from the model. Essentially, we marginalize out the effect of the randomly chosen anchor to obtain the final prediction and the temperature value for a test sample.

Heteroscedastic temperature scaling. Having estimated the temperature for the sample in (1), the heteroscedastic calibration process can be expressed as $H^c(y|\mathrm{x}) = \frac{H(y|\mathrm{x})}{\boldsymbol{\tau}(\mathrm{x})}$. The OOD score for this calibrated sample is given by using any pre-specified scoring function $\mathcal{S}$. In particular, we first obtain class likelihoods from the calibrated logits $P^c(y|\mathrm{x}) = \texttt{SOFTMAX}(H^c(y|\mathrm{x}))$, and compute the negative log likelihood score for OOD detection. We refer to this score as **A**nchor **M**arginalized **P**rediction score (**AMP**). Formally, the AMP scoring function can be defined as:

$$\text{AMP}: \quad \mathcal{S}(\mathrm{x}) = -\frac{1}{N} \sum_{\text{all classes}} \log(\texttt{SOFTMAX}(H^c(y|\mathrm{x}))). \tag{2}$$

For smaller datasets like CIFAR-10/100 the variance tends to be small, and to avoid scaling issues we instead use $H^c(y|\mathrm{x}) = \frac{H(y|\mathrm{x})}{1+\exp(\boldsymbol{\tau}(\mathrm{x}))}$. In Figures 2 and 3 we demonstrate pytorch-like pseudocode for training and inference.

4. Intuition Behind AMP

AMP is effective in distinguishing OOD samples due to the proposed heteroscedastic temperature scaling strategy – by defining the temperature value as a function of the sample, we are able to automatically adjust the scaling to be sensitive to the OOD-ness of the sample. Naturally, this works better than using a single temperature value for the entire dataset. A strong candidate for such a scaling strategy is the *epistemic* uncertainty of a model for a given sample. Intuitively, scaling by uncertainty should improve OOD performance since, by definition, an OOD sample has high epistemic uncertainty, compared to an inlier, which translates to a higher temperature that scales the logits more aggressively. This leads to increased entropy in the resulting prediction probabilities. During inference, AMP estimates the temperature of a sample based on the standard deviation of predictions obtained via multiple anchors. In the following section, we justify why this estimator expressed in (1) is, in fact, related to the epistemic uncertainty.

We utilize neural tangent kernel (NTK) theory (Jacot et al., 2018; Arora et al., 2019; Bietti and Mairal, 2019; Lee et al., 2019), as it provides a convenient framework for analyzing the effect of the modified training proposed in AMP . The basic idea of NTK is that, when the width of a neural network tends to infinity and the learning rate of SGD tends to zero, the function $f(\mathrm{x};\boldsymbol{\theta})$ converges to a solution obtained by kernel regression using the NTK:

$$\mathbf{K}_{\mathrm{x}_i\mathrm{x}_j} = \mathbb{E}_{\boldsymbol{\theta}} \left\langle \frac{\partial f(\mathrm{x}_i,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}, \frac{\partial f(\mathrm{x}_j,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right\rangle. \tag{3}$$

When the samples $\mathrm{x}_i, \mathrm{x}_j \in \mathcal{S}^{d-1}$, i.e., points on the hypersphere and have unit norm, the NTK for a simple 2 layer ReLU MLP can be simplified as a dot product kernel (Arora et al., 2019; Bietti and Mairal, 2019; Lee et al., 2019):

$$\mathbf{K}_{\mathrm{x}_i\mathrm{x}_j} = h_{\mathrm{NTK}}(\mathrm{x}_i^\top \mathrm{x}_j) = \frac{1}{2\pi}\mathrm{x}_i^\top \mathrm{x}_j(\pi - \cos^{-1}(\mathrm{x}_i^\top \mathrm{x}_j)). \tag{4}$$

Let us also consider the prediction on a test sample x_t in the limit as the inner layer widths grow to infinity. It has been shown that (c.f. (Lee et al., 2019; Bietti and Mairal, 2019)):

$$f_\infty(\mathrm{x}_t) = f_0(\mathrm{x}_t) - \mathbf{K}_{\mathrm{x}_t\mathbf{X}}\mathbf{K}_{\mathbf{X}\mathbf{X}}^{-1}(f_0(\mathbf{X}) - \mathbf{Y}), \tag{5}$$

where f_0 is the network with the initial random weights, θ_0, and $\mathbf{X}$ is the matrix of all training data samples.

NTK with neural network anchoring. Recall, the anchoring process involves transforming the input into a tuple: $[\mathrm{c}, \mathrm{x} - \mathrm{c}]$, for a randomly chosen anchor c. In the following, we examine the impact of anchoring on the NTK. Without loss of generality, we assume $[\mathrm{c}, \mathrm{x}_i - \mathrm{c}]$ and $[\mathrm{c}, \mathrm{x}_j - \mathrm{c}]$ are unit norm, so we can simplify $h_{\mathrm{NTK}}([\mathrm{c}, \mathrm{x}_i - \mathrm{c}]^\top [\mathrm{c}, \mathrm{x}_j - \mathrm{c}])$. Further, we use a Taylor series approximation for $\cos^{-1}$: $\cos^{-1}(\mathrm{u} - \mathrm{c}) \approx \cos^{-1}(\mathrm{u}) + \frac{\mathrm{c}}{\sqrt{1-(\mathrm{u}-\mathrm{c})^2}}$.

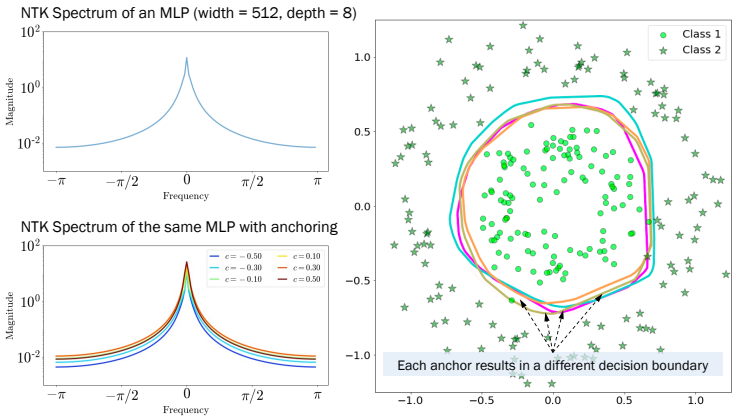


Figure 4: NTK Spectra for a vanilla model and one with neural network anchoring (left). In the simple classification setup on the right, we demonstrate the effect of different anchors on the classifier's predictions. We observe that each anchor produces a slightly different NTK (due to (6)), resulting in meaningful inconsistencies in the classifier's predictions.

Expanding $[\mathrm{c}, \mathrm{x}_i - \mathrm{c}]^\top [\mathrm{c}, \mathrm{x}_j - \mathrm{c}] = \mathrm{x}_i^\top \mathrm{x}_j - \mathrm{c}^\top(\mathrm{x}_i + \mathrm{x}_j - 2\mathrm{c})$ and letting $\mathrm{v} = (\mathrm{x}_i + \mathrm{x}_j - 2\mathrm{c})$, we obtain the expression for h_{NTK} under a shifted domain (from (4)) as follows:

$$\begin{aligned}
\mathbf{K}_{\mathrm{anc}} &= \frac{1}{2\pi}(\mathrm{x}_i^\top \mathrm{x}_j - \mathrm{c}^\top \mathrm{v})(\pi - \cos^{-1}(\mathrm{x}_i^\top \mathrm{x}_j - \mathrm{c}^\top \mathrm{v})) \\
&\approx \frac{1}{2\pi}\mathrm{x}_i^\top \mathrm{x}_j(\pi - \cos^{-1}(\mathrm{x}_i^\top \mathrm{x}_j)) - \frac{1}{2\pi}\mathrm{c}^\top \mathrm{v}(\pi - \cos^{-1}(\mathrm{x}_i^\top \mathrm{x}_j)) - \frac{\mathrm{c}(\mathrm{x}_i^\top \mathrm{x}_j - \mathrm{c}^\top \mathrm{v})}{2\pi\sqrt{1 - (\mathrm{x}_i^\top \mathrm{x}_j - \mathrm{c}^\top \mathrm{v})^2}} \\
&= \mathbf{K}_{\mathrm{x}_i\mathrm{x}_j} - \Gamma_{\mathrm{x}_i,\mathrm{x}_j,\mathrm{c}},
\end{aligned} \tag{6}$$

where we combine all terms dependent on c into $\Gamma_{\mathrm{x}_i,\mathrm{x}_j,\mathrm{c}}$, which also behaves as a dot product kernel. We can see that combining this *stochastic* NTK (in c) into (5), the prediction on a test sample is correspondingly stochastic under a fixed initialization θ_0. In other words, neural network anchoring effectively perturbs the hypothesis space of the neural network resulting in an ensembling-like behavior, that effectively produces different predictions, conditioned on the anchor choice. We illustrate the process in Figure 1, and compute the NTK spectrum (as done in Tancik et al.) along with a demonstrative toy classification problem in Fig. 4. Our training objective (with randomly chosen anchors) forces the predictions on a sample across anchors to be the same for in distribution samples, and therefore the disagreement (measured by sum of standard deviation, (1)) is larger when the sample is OOD. This is similar, in principle, to deep ensembles (Ovadia et al., 2019), that measure prediction variance by training multiple models with different random initializations $\boldsymbol{\theta}_0$.

5. Experiments and Results

In this section we evaluate the proposed anchoring-based OOD detection on a wide variety of OOD benchmarks, with several different model architectures, and OOD settings.

Setup. Our modification to any standard neural network architecture is minimal – we only change the first convolutional layer to accept a 6 channel input (3 channels for anchor and residual each) instead of the original 3 channels. We use standard hyper-parameters to train all our models. For our experiments with CIFAR10/100 (Krizhevsky et al., 2009) datasets, we trained a wide ResNet (WRN) model for 200 epochs, with an initial learning rate of 0.1, and a decay of 0.2 at 60, 120, and 160 epochs. In addition to WRN, we also experimented with ResNet-18 and ResNet-34 architectures. For our ImageNet (Russakovsky et al., 2015) experiments we trained a ResNet-50 model with an initial learning rate of 0.1 and a decay of 0.1 every 30 epochs. Note that, we used the same normalization scheme for pre-processing both in- and out- distribution data sets (at train and test times). As discussed in the previous section, for our method, we used a single randomly chosen anchor for every input image in a mini-batch. From our extensive empirical studies, we observed no significant difference in the top-1 accuracy of the anchored models on any of the benchmarks. For example, with the WRN-40-2 on CIFAR-10, we obtained an accuracy of 95.1 on average, and on CIFAR-100 a top-1 accuracy of 76.1. On ImageNet, our ResNet-50 model trained for 120 epochs resulted in a top-1 accuracy of 76.0. More details of the benchmarks and the exact hyper-parameter settings are provided in the supplement.

Baselines. We performed comparisons with several widely-adopted OOD detection approaches: (a) Maximum Softmax Probability (MSP); (b) ODIN (Liang et al., 2017); (c) Energy (Liu et al., 2020); (d) Gram Matrices GM (Sastry and Oore, 2020) that uses latent space deviation to detect OOD, but has been shown to perform better than scoring functions such as Mahalanobis detection (Lee et al., 2018b) without requiring exposure to outlier data. We also evaluated our approach against uncertainty-based OOD detectors for the ImageNet experiments, namely MC-dropout (Gal and Ghahramani, 2016), deep ensembles (Lakshminarayanan et al., 2017) and techniques that require post-hoc calibration on a validation set like temperature scaling (Guo et al., 2017), SVI (Blundell et al., 2015), and AVuC (Krishnan and Tickoo, 2020).

Metrics: We used 4 standard metrics to evaluate OOD detection performance across our benchmarks – (a) **FPR95**: False positive rate of examples from the OOD set when the true positive rate (TPR) of the in-distribution is set at 95%; (b) **AUROC**: Area under the receiver operating characteristic curve; (3) **AUPR**: Area under the precision-recall curve for both In/Out sets depending on which one is considered positive; (4) **DTACC**: Detection accuracy measures the maximum possible OOD detection accuracy across all thresholds as proposed in (Lee et al., 2018b).

	OOD	FPR95 ↓	AUROC ↑	AUPR ↑
		MSP / ODIN / Energy / AMP (ours)		
CIFAR-10	iSUN	56.03 / 32.05 / 33.68 / **16.59**	89.83 / 93.50 / 92.62 / **97.16**	97.74 / **98.54** / 98.27 / 97.83
	LSUN (R)	52.15 / 26.62 / 27.58 / **13.73**	91.37 / 94.57 / 94.24 / **97.77**	98.12 / **98.77** / 98.67 / 97.77
	LSUN (C)	30.80 / 15.52 / 8.26 / **1.50**	95.65 / 97.04 / 98.35 / **99.55**	99.13 / 99.33 / **99.66** / 99.57
	Places365	59.48 / 57.40 / 40.14 / **19.89**	88.20 / 84.49 / 89.89 / **95.79**	97.10 / 95.82 / **97.30** / 87.28
	Texture	59.28 / 49.12 / 52.79 / **35.43**	88.50 / 84.97 / 85.22 / **93.61**	**97.16** / 95.28 / 95.41 / 96.14
	SVHN	48.49 / 33.55 / 35.59 / **5.19**	91.89 / 91.96 / 90.96 / **98.10**	**98.27** / 98.00 / 97.64 / 97.51
	Average	42.71 / 35.71 / 33.01 / **15.39**	90.91 / 91.01 / 91.88 / **96.99**	**97.91** / 97.62 / 97.82 / 96.02
CIFAR-100	iSUN	82.80 / 68.51 / 81.10 / **67.15**	75.46 / 82.69 / 78.91 / **83.79**	94.06 / **95.80** /94.91 / 85.84
	LSUN (R)	82.42 / 71.96 / 79.47 / **61.73**	75.38 / 81.82 / 79.23 / **85.64**	94.06 / **95.65** / 94.96 / 84.30
	LSUN (C)	66.54 / 55.55 / 35.32 / **4.16**	83.79 / 87.73 / 93.53 / **99.18**	96.35 / 97.22 / 98.62 / **99.19**
	Places365	82.84 / 87.88 / 80.56 / **65.18**	73.78 / 71.63 / 75.44 / **85.78**	93.29 / 92.56 / **93.45** / 69.65
	Texture	83.29 / **79.27** / 79.41 / 81.81	73.34 / 73.45 / **76.28** / 71.36	92.89 / 92.75 / **93.63** / 79.72
	SVHN	84.59 / 84.66 / 85.82 / **12.57**	71.44 / 67.26 / 73.99 / **97.85**	92.93 / 91.38 / 93.65 / **95.25**
	Average	80.41 / 74.64 / 73.61 / **48.77**	75.53 / 77.43 / 79.56 / **87.27**	93.93 / 94.23 / **94.87** / 85.66

Table 2: **OOD Detection with WideResNet-40-2:** We compare with a range of different OOD detection methods on the commonly used OOD benchmark. We see a consistent improvement in performance when using AMP , over existing baselines – across both CIFAR-10/100 models. AMP is particularly good in suppressing false positives as reflected by the FPR95 metric.

5.1. Performance on OOD Benchmarks

We begin by evaluating AMP on a commonly used OOD detection benchmark first introduced by Liang *et al.* (Liang et al., 2017). In this experiment, predictive models were trained with CIFAR-10 or CIFAR-100 as the in-distribution, and then used to detect out-of-distribution data chosen from one of the six datasets (details in the supplement). We follow the experimental protocol from (Liu et al., 2020), and used the WideResNet architecture WRN-40-2 (Zagoruyko and Komodakis, 2016). In Table 2, we show results across all the six datasets for both CIFAR-10/100, using the three metrics. On average, we find that AMP significantly improves on the challenging FPR95 metric (nearly 50% lower than the next best), while also providing substantial gains in the AUROC metric. We notice that, these gains persist even with CIFAR-100, which is known to be a much harder setting (as reflected by the higher false positive rates). We do not assume access to additional OOD data (for model finetuning), unlike both energy- and ODIN-based detection that perform similarly, while our approach significantly improves upon them ($\approx 25\%$ drop in the FPR metric).

	Method	FPR95 ↓	AUROC ↑	DTACC ↑
CIFAR-10	MSP	55.12	90.37	84.65
	ODIN	33.40	92.70	85.28
	Energy	28.89	94.47	88.58
	Mahal.*	14.8	**97.33**	**93.15**
	GM	14.83	96.33	92.14
	AMP (ours)	**12.33**	**97.20**	92.84
CIFAR-100	MSP	80.22	77.22	70.95
	ODIN	60.82	85.41	78.61
	Energy	70.96	82.44	75.67
	Mahal.*	**24.36**	**94.07**	**88.51**
	GM	49.87	89.96	83.50
	AMP (ours)	49.70	89.97	82.50

Table 3: **OOD detection performance with ResNet-34:**. Averaged across 7 datasets following Sastry and Oore (2020). Here, 'Mahal.' indicates the Mahalanobis score, which uses additional outlier data during training.

	Method	ResNet-34 FPR95 ↓ / AUROC ↑
C10 → C100	ODIN	58.0 / 88.2
	Energy	47.5 / 88.4
	GM	59.8 / 83.6
	Mahal.*	58.4 / 88.2
	AMP (ours)	**43.5 / 90.2**
C100 → C10	ODIN	81.3 / 77.2
	Energy	80.9 / 77.0
	GM	83.1 / 74.5
	Mahal.*	**79.8** / 77.5
	AMP	82.5 / **79.9**

Table 4: AMP is effective on near OOD detection with CIFAR-10↔100 respectively.

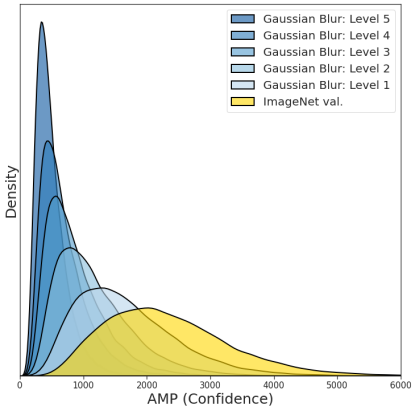
Method	AUROC ↑	DTACC ↑	AUPR-in/out ↑
ResNet-50 (He et al., 2016)	93.36	86.08	92.82 / 93.71
Temp-Scal (Guo et al., 2017)	93.71	86.47	93.21 / 94.01
Deep Ens (Lakshminarayanan et al., 2017)	95.49	88.82	95.31 / 95.64
MCD (Gal and Ghahramani, 2016)	96.38	89.98	96.16 / 96.67
SVI (Blundell et al., 2015)	96.40	90.03	95.97 / 96.83
SVI-AvUC (Krishnan and Tickoo, 2020)	97.60	92.07	97.39 / 97.85
AMP (ours)	**99.07**	**95.14**	**99.10** / **99.08**

Table 5: **UQ-based detection of ImageNet-C with ResNet-50:** We evaluate AMP on the benchmark introduced by (Krishnan and Tickoo, 2020) where we use Gaussian Blur of level 5 intensity as the OOD, and the clean ImageNet validation as the ID. For comparisons with OOD methods on this benchmark, refer to Figure 5(*b*).

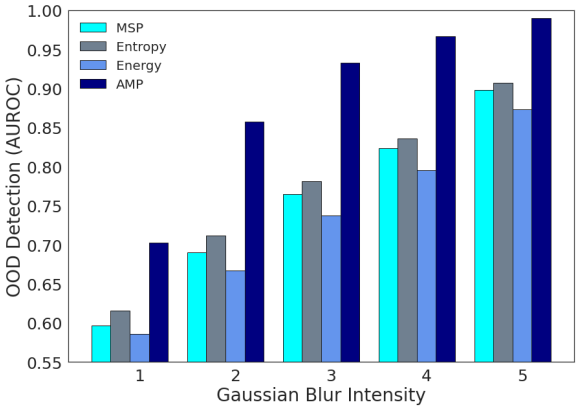
Next, in Table 3, we evaluate on the benchmark introduced in (Sastry and Oore, 2020) using ResNet-34. This consists of 7 different OOD datasets – iSUN, LSUN (R), LSUN (C), TinyImageNet (R), TinyImageNet (C), CIFAR10/100, SVHN. We observe that AMP comes second to Mahalanobis on CIFAR-100 while matching it on CIFAR-10, even though we do not use outlier data for fine-tuning. In order to make a fair comparison on the same OOD samples for all datasets, we use GM without the validation data (and corresponding normalization) – the performance improvements were consistently observed in most cases. Detailed results for this benchmark are available in the supplement.

Uncertainty based ImageNet-OOD detection. The prediction variance from AMP can be interpreted as a measure of unreliability in the model's predictions. Hence, a natural evaluation is to leverage this unreliability estimate as a score for OOD detection, since it is expected to be high as we move away from the original data manifold (*i.e.*, OOD), while being low for in-distribution data. We analyzed this performance on ImageNet-C data (Hendrycks

and Dietterich, 2019) using a ResNet-50 architecture trained on ImageNet-1K, as shown in Table 5. Specifically, we used the OOD dataset obtained via Gaussian blur corruption at intensity 5. We find that, by leveraging our unreliability score to perform adaptive temperature scaling, AMP is highly effective for OOD detection. In fact, this improves over several state-of-the-art uncertainty estimators on this benchmark.



(*a*) Densities of scores between in and out distribution ImageNet-C.



(*b*) AUROC (↑) across corruption levels for AMP

Figure 5: **From near to far OOD:** AMP outperforms baselines consistently at all the intensity levels (Fig. 5(*b*)), and produces meaningful scores (Fig. 5(*a*)) when moving from near (level 1) to far (level 5) OOD sets.

Near OOD detection. Here, we consider the more challenging "near OOD" detection task that seeks to separate CIFAR-10 and CIFAR-100 datasets, by using a model trained on one of them while treating the other as the OOD set. Since both are drawn from the same image distribution but have mutually exclusive classes, this is extremely challenging and often causes OOD detectors to fail. We measure OOD performance using both FPR and AUROC as shown in Table 4. We see that methods like Mahalanobis detector (Lee et al., 2018b), Gram Matrices (Sastry and Oore, 2020), which are otherwise very competitive tends to fail in this task – whereas AMP is able to separate them better, indicating its effectiveness for both near and far OOD tasks.

We also study how AMP performs as the OOD set is artificially made to go farther from the ImageNet validation set using the Gaussian blur corruption of varying intensity. We expect that an effective scoring mechanism must reflect this well in its scores. We demonstrate this in Fig. 5(*a*) that shows the kernel density plot of AMP scores for both in- and out-distribution datasets across 5 intensity levels. Since the score is designed so as to be higher for the in-distribution set, we find a gradual and meaningful transition between the clean in-distribution set to the farthest OOD set (tight in the low confidence area). This is reflected in the OOD detection performance as measured by AUROC shown in Fig. 5(*b*), where all three variants of scoring functions with AMP outperform competing approaches consistently across different corruption levels.

In-distribution	Method	Needs OOD Exposure?	FPR95 ↓	AUROC ↑	AUPR(In/Out) ↑
CIFAR-10 (ResNet-18)	ODIN (Liang et al., 2017)	✗	52.00	82.00	73.13 / 85.12
	Energy (Liu et al., 2020)	✗	50.03	83.83	77.15 / 85.11
	OE (Hendrycks et al., 2019)	✓	50.53	88.93	87.55 / 87.83
	MCD (Yu and Aizawa, 2019)	✓	73.02	83.89	83.39 / 80.53
	UDG (Yang et al., 2021)	✓	**36.22**	**93.78**	**93.61** / **92.61**
	AMP	✗	**36.82**	92.40	91.23 / 90.91
CIFAR-100 (ResNet-18)	ODIN (Liang et al., 2017)	✗	81.89	77.98	78.54 / 72.56
	Energy (Liu et al., 2020)	✗	81.66	79.31	80.54 / 72.82
	OE (Hendrycks et al., 2019)	✓	80.06	78.46	80.22 / 71.83
	MCD (Yu and Aizawa, 2019)	✓	85.14	74.82	75.93 / 69.14
	UDG (Yang et al., 2021)	✓	75.45	79.63	80.69 / 74.10
	AMP	✗	**70.34**	**82.22**	**84.14** / **76.20**

Table 6: **SCOOD benchmark with ResNet-18.** AMP outperforms even the best performing techniques on the recent semantically coherent OOD bencmark (Yang et al., 2021) on both CIFAR-10 and CIFAR-100 inspite of not requiring outlier exposure. The methods OE (Hendrycks et al., 2019), MCD (Yu and Aizawa, 2019), and UDG use Tiny-ImageNet during training.

5.2. Semantically Coherent OOD (SCOOD)

The current suite of OOD detection benchmarks rely on separating the dataset on which a model is trained from another dataset marked as OOD, for example, CIFAR-SVHN. However, considering that dataset specific biases are prominent, it is likely that many of the OOD detection algorithms tend to overemphasize dataset-specific *noise* in metrics such as FPR or AUROC. Further, it is likely there are factors other than semantic content that are contributing to very high OOD performance. To address this, we consider the recently proposed SCOOD (semantically coherent OOD detection) benchmark (Yang et al., 2021), which effectively re-samples the in- and out-distribution datasets such that the only truly distinguishing factor between the two is the semantic content. This also includes transferring semantically similar images from the OOD dataset into the in-distribution set (e.g., cats from TinyImageNet into CIFAR-100), and getting rid of resizing artifacts so that the OOD performance reflects the true performance.

In Table 6 we report average OOD detection performance with AMP on the SCOOD benchmark, which is comprised of 6 different resampled/resized datasets – Texture, SVHN, CIFAR-10/100, Tiny-ImageNet, LSUN, Places365. We trained ResNet-18 models on modified CIFAR-10/100 training sets provided by SCOOD (Yang et al., 2021) and test on their custom OOD sets for fair comparison. We compare the different methods on FPR95, AUROC, and AUPR (In/Out) metrics. The current best performing approach on SCOOD, UDG (Yang et al., 2021) additionally also uses outlier data from TinyImageNet for training, similar to other approaches like OE. We find that AMP is the best performing method on CIFAR-100 across all the metrics, while on CIFAR-10 AMP performs comparably to UDG in terms of FPR95, while being a close second on the other metrics – in spite of not having any access to outlier data, while being significantly better than other comparable baselines.

In Distribution	Method	Pillow Resizing from original LSUN				Average
		`nearest`*	`bilinear`	`bicubic`	`lanczos`	
CIFAR-10 (ResNet-34)	MSP	41.5 / 94.0	47.8 / 91.6	45.5 / 92.2	45.3 / 92.4	46.9 / 92.2
	Energy	28.6 / 98.4	34.5 / 92.9	33.0 / 93.4	32.0 / 93.8	30.4 / 94.1
	GM	**1.8 / 99.2**	46.2 / 90.7	49.0 / 90.6	46.3 / 91.3	25.6 / 94.9
	AMP	7.1 / 98.4	**13.0 / 97.4**	**13.9 / 97.2**	**14.3 / 97.2**	**9.6 / 98.1**
CIFAR-100 (ResNet-34)	MSP	69.0 / 83.8	80.0 / 79.6	84.0 / 79.9	84.0 / 80.5	79.9 / 79.2
	Energy	52.4 / 89.3	84.9 / 74.9	83.3 / 75.7	81.4 / 76.8	73.9 / 79.6
	GM	**38.7 / 94.4**	78.1 / 79.0	77.6 / 80.0	76.0 / 81.5	60.9 / 86.5
	AMP	51.1 / 90.5	**75.4 / 81.2**	**74.0 / 81.7**	**71.8 / 82.7**	**58.3 / 86.9**

*aliasing artifacts are likely Performance measures are FPR95 ↓ / AUROC ↑

Table 7: **Resizing methods vs OOD benchmarks:** We study the effect of resizing on OOD performance. Here we use ResNet-34 trained on CIFAR-100 as our "in" dataset, to detect 10K LSUN test images. These are resized using different interpolation algorithms from the Pillow package. We evaluate the performance of different OOD detection algorithms on these various datasets, and observe that AMP consistently performs well across all the variants of LSUN.

5.3. Robustness to Resizing Artifacts

As stated previously, OOD detection benchmarks can be hard to interpret when OOD datasets are laden with their dataset-specific noise/biases, often reflecting in overly optimistic OOD performance. A key source for such kinds of noise is artifacts obtained from resizing – since most datasets have different sized images, they are resized typically using a nearest neighbor interpolation algorithm before running any OOD detection algorithm. The issue with resizing packages like OpenCV, or native resizing in Pytorch or Tensorflow was recently studied in detail in (Parmar et al., 2021) with its impact on FID scores for generative models – they concluded that the Pillow resizing with a `bicubic` interpolation scheme was the most reliable and gets rid of aliasing artifacts the best. Designing specific resizing frameworks has also been shown to have an impact on classification accuracy (Talebi and Milanfar, 2021).

This issue is further exaggerated in the current OOD setup relying on CIFAR-10/100 because the images are very small (32×32) when compared to other datasets. Specifically for OOD, this was pointed out in the SCOOD benchmark (Yang et al., 2021), as being one of the reasons to re-design these benchmarks from scratch, however there is no ablation on how resizing alone actually affects performance. Here, we evaluate how some of the best OOD detection algorithms perform on the same in-out distribution experiment – CIFAR-10/100 $\rightarrow$ LSUN. Since the original LSUN images are of a much larger size, we resize them using the Pillow resizing library using different types of interpolations.

We report the results of this study in Table 7. The first striking observation is the amount of variance in OOD performance across the different LSUN variants – across the baselines, and metrics we used. In particular – the case of `nearest` interpolation is the most similar to the LSUN (R) benchmark , and it is expected to introduce the most amount of aliasing artifacts, and existing approach produce unusually low FPR scores. However, the results the more sophisticated interpolation methods are expected to be more indicative of the true OOD performance, since they are not prone to aliasing. Interestingly, we note

that AMP performs the best on all of these cases. Our trend across datasets is similar to energy-based OOD (Liu et al., 2020) and on average, AMP significantly outperforms the other baselines on both the metrics considered.

5.4. Ablation Studies

Finally, we study the two important aspects of AMP here and their impact on OOD performance using the WRN-40-2 on the CIFAR-10/SVHN benchmark. In Table 8, we report FPR95 as our metric of choice since its the most sensitive and reflective of the performance. We ablate on two factors – the type of transformation used during training to anchor the neural network, and the number of anchors needed during inference time. We see that simple functions like Gaussian blur or color jitter drastically improve the performance. In all our experiments, we used a combination of all the five corruptions, with 5 anchors. As can be seen, there is not a significant difference in performance while increasing the number of anchors, but a big boost in using consistency training using any of the corruption functions.

Corruption \ # Anchors	**2**	**5**	**10**	**20**
None (trivial)	50.71	50.73	50.85	50.89
+ `ColorJitter` + `GaussianBlur`	6.53	6.31	6.36	6.28
+ `HorizontalFlip` + `Grayscale`	9.96	9.85	10.11	9.89
+ `ResizedCrop`	5.32	5.19	5.35	5.22

Table 8: **Ablation studies** demonstrating FPR95 for CIFAR-10-SVHN OOD experiment using types of transformations and varying number of anchors. All these transformations are applied at random every time they are executed.

6. Discussion

In this paper, we introduced *anchoring* as a strategy to achieve effective heterscedastic temperature scaling for state-of-the-art OOD detection on a large suite of benchmarks including near and semantically coherent-OOD problems. Using NTK theory, we show that our temperature estimates are closely linked to epistemic uncertainty of the classifier, explaining it superior performance. We also introduced a new benchmark that evaluates the robustness of OOD detection methods against resizing artifacts. Anchoring is a powerful new mechanism to estimate confidence or reliability of a model, and shows promise well beyond OOD itself including studying its properties as an uncertainty estimator in more general settings.

Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. Supported by the LDRD Program under projects 21-ERD-028, 22-ERD-006 and released under LLNL-JRNL-829478.

References

Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pages 322–332. PMLR, 2019.

Alberto Bietti and Julien Mairal. On the inductive bias of neural tangent kernels. *Advances in Neural Information Processing Systems*, 32, 2019.

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.

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.

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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJz6tiCqYm.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *ICLR*, 2019.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Moksh Jain, Salem Lahlou, Hadi Nekoei, Victor Butoi, Paul Bertin, Jarrid Rector-Brooks, Maksym Korablyov, and Yoshua Bengio. Deup: Direct epistemic uncertainty prediction. *arXiv preprint arXiv:2102.08501*, 2021.

Ranganath Krishnan and Omesh Tickoo. Improving model calibration with accuracy versus uncertainty optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18237–18248. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/d3d9446802a44259755d38e6d163e820-Paper.pdf.

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. *Advances in neural information processing systems*, 30, 2017.

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32, 2019.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018a.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, 2018b.

Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, 2017.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *NeurIPS*, 2020.

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On buggy resizing libraries and surprising subtleties in fid calculation. *arXiv preprint arXiv:2104.11222*, 2021.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Chandramouli Shama Sastry and Sageev Oore. Detecting out-of-distribution examples with gram matrices. In *International Conference on Machine Learning*, pages 8491–8501. PMLR, 2020.

Hossein Talebi and Peyman Milanfar. Learning to resize images for computer vision tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 497–506, 2021.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020.

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.

Jingkang Yang, Haoqi Wang, Litong Feng, Xiaopeng Yan, Huabin Zheng, Wayne Zhang, and Ziwei Liu. Semantically coherent out-of-distribution detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8301–8309, 2021.

Qing Yu and Kiyoharu Aizawa. Unsupervised out-of-distribution detection by maximum classifier discrepancy. In *ICCV*, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.