

---

# A Simple Fix to Mahalanobis Distance for Improving Near-OOD Detection

---

Jie Ren<sup>1</sup> Stanislav Fort<sup>\*2</sup> Jeremiah Liu<sup>\*13</sup> Abhijit Guha Roy<sup>4</sup> Shreyas Padhy<sup>1</sup> Balaji Lakshminarayanan<sup>1</sup>

## Abstract

Mahalanobis distance (MD) is a simple and popular post-processing method for detecting out-of-distribution (OOD) inputs in neural networks. We analyze its failure modes for near-OOD detection and propose a simple fix called *relative Mahalanobis distance* (RMD) which improves performance and is more robust to hyperparameter choice. On a wide selection of challenging vision, language, and biology OOD benchmarks (CIFAR-100 vs CIFAR-10, CLINC OOD intent detection, Genomics OOD), we show that RMD meaningfully improves upon MD performance (by up to 15% AUROC on genomics OOD).

## 1 Introduction

Out-of-distribution (OOD) detection is critical for deploying machine learning models in safety critical applications [1]. A lot of progress has been made in improving OOD detection by training complicated generative models [2; 15; 20; 14], modifying objective functions [22], and exposing to OOD samples while training [8]. Although such methods have promising results, they might require training and deploying a separate model in addition to the classifier, or rely on OOD data for training and/or hyperparameter selection, which are not available in some applications. A Mahalanobis distance (MD) based OOD detection method [12] is a simpler approach which is easy to use. This method does not involve re-training the model and works out-of-the-box for any trained model. MD is a popular approach due to its simplicity.

Although MD based methods are highly effective in identifying *far* OOD samples (samples which are semantically and stylistically very different from the in-distribution samples, e.g., CIFAR-10 vs. SVHN), we identify that it often fails for *near* OOD samples (samples which are semantically similar to the in-distribution samples [21], e.g.,

CIFAR-100 vs. CIFAR-10) that are more challenging to detect. In this paper, we focus primarily on the near OOD detection task and investigate why the MD method fails in these cases. We propose *relative Mahalanobis distance* (RMD), a simple fix to the MD, and demonstrate its effectiveness in multiple near-OOD tasks. Our solution is as simple to use as MD, and it does not involve any complicated re-training or training OOD data.

## 2 Methods

In this section, we briefly review the Mahalanobis distance method and introduce our proposed modifications to make it effective for near-OOD detection tasks.

**Mahalanobis distance based OOD detection** The Mahalanobis distance (MD) [12] method uses intermediate feature maps of a trained deep neural network. The most common choice for the feature map is the output of the penultimate layer just before the classification layer. Let us indicate this feature map as  $\mathbf{z}_i = f(\mathbf{x}_i)$  for an input  $\mathbf{x}_i$ . For an in-distribution dataset with  $K$  unique classes, MD method fits  $K$  class conditional Gaussian distributions  $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$ ,  $k = 1, 2, \dots, K$  to each of the  $K$  in-distribution classes based on the feature maps  $\mathbf{z}_i$ . We estimate the mean vectors and covariance matrix as:  $\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i: y_i=k} \mathbf{z}_i$ , for  $k = 1, \dots, K$ , and  $\boldsymbol{\Sigma} = \frac{1}{N} \sum_{k=1}^K \sum_{i: y_i=k} (\mathbf{z}_i - \boldsymbol{\mu}_k)(\mathbf{z}_i - \boldsymbol{\mu}_k)^T$ . Note that class-conditional means  $\boldsymbol{\mu}_k$  are independent for each classes, while the covariance matrix  $\boldsymbol{\Sigma}$  is shared by all classes to avoid under-fitting errors.

For a test input  $\mathbf{x}'$ , the method computes the Mahalanobis distances from the feature map of a test input  $\mathbf{z}' = f(\mathbf{x}')$  to each of the fitted  $K$  in-distribution Gaussian distributions  $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$ ,  $k \in \{1, \dots, K\}$  given by  $\text{MD}_k(\mathbf{z}')$ . The minimum of the distances over all classes indicates the uncertainty score  $\mathcal{U}(\mathbf{x}')$  and its negative indicates the confidence score  $\mathcal{C}(\mathbf{x}') = -\mathcal{U}(\mathbf{x}')$ . These are computed as

$$\text{MD}_k(\mathbf{z}') = (\mathbf{z}' - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}^{-1} (\mathbf{z}' - \boldsymbol{\mu}_k), \quad (1)$$

$$\mathcal{C}(\mathbf{x}') = -\min_k \{\text{MD}_k(\mathbf{z}')\}. \quad (2)$$

This confidence score is used as a signal to classify a test input  $\mathbf{x}'$  as an in-distribution or OOD sample.

\*Work done at Google Research. <sup>1</sup>Google Research <sup>2</sup>Stanford University <sup>3</sup>Harvard University <sup>4</sup>Google Health. Correspondence to: Jie Ren <jjren@google.com>, Balaji Lakshminarayanan <balajiln@google.com>.

**Our proposed Relative Mahalanobis distance** As we will demonstrate in Sec. 3 and Appendix. D, OOD detection performance using MD degrades for near-OOD scenarios. We draw our inspiration from the prior work by Ren et al. [20] showing that the raw density from deep generative models may fail at OOD detection and proposing to fix this using a likelihood ratio between two generative models (one modeling the sophisticated foreground distribution and the other modeling the background distribution) as confidence score. Similarly, we propose Relative Mahalanobis distance (RMD) defined as

$$\text{RMD}_k(\mathbf{z}') = \text{MD}_k(\mathbf{z}') - \text{MD}_0(\mathbf{z}').$$

Here,  $\text{MD}_0(\mathbf{z}')$  indicates the Mahalanobis distance of sample  $\mathbf{z}'$  to a distribution fitted to the entire training data not considering the class labels:  $\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ , where  $\boldsymbol{\mu}_0 = \frac{1}{N} \sum_{i=1}^N \mathbf{z}_i$  and  $\boldsymbol{\Sigma}_0 = \frac{1}{N} \sum_{i=1}^N (\mathbf{z}_i - \boldsymbol{\mu}_0)(\mathbf{z}_i - \boldsymbol{\mu}_0)^T$ . This is a good proxy for the background distribution. The confidence score using RMD is given by

$$C^{\text{RMD}}(\mathbf{x}') = -\min_k \{\text{RMD}_k(\mathbf{z}')\}. \quad (3)$$

See Appendix A for the pseudocode.

RMD is equivalent to computing a likelihood ratio  $\max_k (\log p_k(\mathbf{z}') - \log p_0(\mathbf{z}'))$ , where  $p_k$  is a Gaussian fit using class-specific data and  $p_0$  is a Gaussian fit using data from all classes. Note that this can easily be extended to the case where  $p_k$  and  $p_0$  are represented by more powerful generative models such as flows [16; 17].

Previous literature [9] discussed a similar topic however their work mainly focused on far-OOD, and their proposed method called Partial Mahalanobis distance (PMD) required a hyper-parameter (number of eigen-bases to consider), while our method performs better for near-OOD and is hyper-parameter free. See Appendix C for the comparison of PMD and RMD.

### 3 Failure Modes of Mahalanobis distance

To better understand the failure mode of Mahalanobis distance and to visualize its difference from the Relative Mahalanobis, we perform an eigen-analysis to understand how these methods weight each dimension [9]. Specifically, we rewrite the Mahalanobis distance using eigenvectors  $\mathbf{v}_d$  of the covariance matrix  $\boldsymbol{\Sigma}$  as  $\text{MD}(\mathbf{z}') = (\mathbf{z}' - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{z}' - \boldsymbol{\mu}) = \sum_{d=1}^D l_d^2 / \lambda_d$ , where  $D$  is the dimension of the feature map,  $\lambda_d$  is the  $d^{\text{th}}$  eigenvalue, and  $l_d = \mathbf{v}_d^T (\mathbf{z}' - \boldsymbol{\mu})$  is the projected coordinate of  $(\mathbf{z}' - \boldsymbol{\mu})$  to the  $d^{\text{th}}$  eigen-basis  $\mathbf{v}_d$  such that  $l_d^2 / \lambda_d$  can be regarded as the 1D Mahalanobis distance from the projected coordinate to the 1D Gaussian distribution  $\mathcal{N}(0, \lambda_d)$ . The  $D$  eigen-bases are independent of each other.

In the CIFAR-100 vs CIFAR-10 experiment, we found that OOD inputs have significantly greater mean distance (i.e.

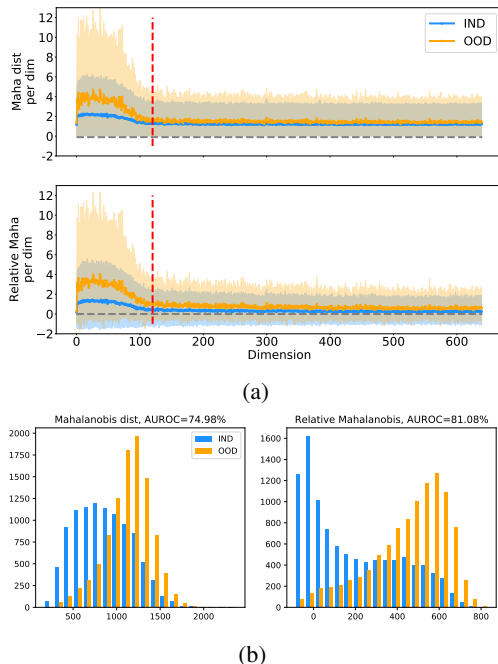


Figure 1: (a) Mahalanobis distance (top) and Relative Mahalanobis Distance (bottom) to CIFAR-100 (IND) and CIFAR-10 (OOD) along the  $d^{\text{th}}$  eigen-basis. The solid lines represent the means over the IND and OOD test data respectively. The shading indicates the [10%, 90%] quantiles. The 120 top dimensions (before the red threshold) have distinct Mahalanobis distance between IND and OOD, while the later dimensions have similar Mahalanobis distances between IND and OOD, confounding the final score. (b) Histograms of the Mahalanobis distance and Relative Mahalanobis distance for IND and OOD.

the average distance over the test samples) in the top 120 dimensions with the largest eigenvalues, while in the remaining dimensions the OOD inputs have similar mean distance with the IND inputs (see Figure 1a, top). Since the final Mahalanobis distance is the sum of the distance per dimension (this can be visualized as the area under the curve in Figure 1a), we see that the later dimensions contribute a significant portion to the final score, overwhelming the top dimensions and making it harder to distinguish OOD from IND (AUROC=74.98%).

Next we fit a class-independent 1D Gaussian as the background model in each dimension and compute RMD per dimension. As shown in Figure 1a (bottom), using RMD, the contributions of the later dimensions are significantly reduced to nearly zero, while the top dimensions still provide a good distinction between IND and OOD. As a result, the AUROC using RMD is improved to 81.08%.

We conjecture that the first 120 dimensions are discriminative features that contain different semantic meanings for different IND classes and OOD, while the remaining di-

mensions are the common features shared by the IND and OOD. To support our conjecture, we simulated a simple dataset following a high-dimensional Gaussian with a diagonal covariance matrix and different means for different classes. In particular, we set IND and OOD to have distinct means in the first dimension (discriminative feature) and the same mean in the remaining dimensions (non-discriminative features). Since MD is the sum over all the dimensions, the sum along those non-discriminative dimensions can overwhelm that of the discriminative dimension. As a result, the AUROC is only 83.13%. Using RMD, we remove the effect of the non-discriminative dimensions as for those dimensions the estimated  $\mathcal{N}(\mu_k, \Sigma) \approx \mathcal{N}(\mu_0, \Sigma_0)$ , detecting OOD perfectly with 100% AUROC using the RMD.

## 4 Experiments and Results

As indicated in the previous section, in this work we primarily focus on near-OOD detection tasks. We choose the following established near-OOD setups: (i) CIFAR-100 vs. CIFAR-10, (ii) CIFAR-10 vs. CIFAR-100, (iii) Genomics OOD benchmark [20] and (iv) CLINC Intent OOD benchmark [11; 13]. As baselines, we compare our proposed RMD to traditional MD and maximum of softmax probability (MSP) [6], both working directly with out-of-the-box trained models. Note that most OOD detection methods require re-training of the models and complicated hyperparameter tuning, which we do not consider for comparison. We also ablate over different choices of model architectures with and without large scale pre-trained networks. The results are presented in the following sections.

### 4.1 Models without pre-training

In this section, we train our models from scratch using the in-distribution data. For CIFAR-10/100 tasks we use a Wide ResNet 28-10 architecture as the backbone. For genomics OOD benchmark we use a 1D CNN architecture consistent with [20]. For all benchmarks, at the end of training, we extract the feature maps for test IND and OOD inputs, and evaluate the OOD performance for our proposed RMD and compare it with MD and MSP. As seen in Table 1, contrasting MD and RMD, we observe a consistent improvement in AUROC for all benchmarks with gains ranging from 1.2 points to 15.8 points. Comparing RMD to MSP, we observe a significant gain of 2.5 points for the Genomics OOD benchmark and partial gains for CIFAR-10/100 benchmarks. This substantiates our claim that our proposed RMD boosts near-OOD detection performance.

**Using flows for  $p_0$  and  $p_k$**  To demonstrate that our proposed idea can be extended to more powerful density models, we fit the feature maps using a one-layer masked auto-regressive flow [16] for the CIFAR-100 vs CIFAR-10 benchmark. The AUROCs for using  $\max_k \log p_k^{\text{flow}}(z')$

Benchmark	MD	RMD	MSP
CIFAR-100 vs CIFAR-10	74.91%	<b>81.01%</b>	80.14%
CIFAR-10 vs CIFAR-100	88.49%	<b>89.71%</b>	89.27%
Genomics OOD	53.10% <sup>1</sup>	<b>68.98%</b>	66.53%

Table 1: Comparison of OOD-AUROC on the near-OOD benchmarks.

and  $\max_k (\log p_k^{\text{flow}}(z') - \log p_0^{\text{flow}}(z'))$  are 76.10%, and 78.34% respectively, showing that our proposal works for non-Gaussian density models as well.

### 4.2 Models with pre-training

Massive models pre-trained on large scale datasets are becoming a standard practice in modern image recognition and language classification tasks. It has been shown that the high-quality features learnt during this pre-training stage can be very useful in boosting the performance of the downstream task [7; 18; 5]. In this section, we investigate if such high-quality representations also aid in better OOD detection and how our proposed RMD performs in such a setting, using different pre-trained models as architectural backbone for OOD detection. Specifically, we consider Vision Transformer (ViT) [4], Big Transfer (BiT) [10], and CLIP [19] for CIFAR-10/100 benchmarks, and the unsupervised BERT style pre-training model [3] for genomics<sup>2</sup> and CLINC benchmarks.

We investigate two settings: (i) directly using pre-trained models for OOD detection and (ii) fine-tuning the pre-trained model on the in-distribution dataset for OOD detection.

**Pre-trained models without fine-tuning** We present our results in Table 2, comparing MD and RMD for all benchmarks using different pre-trained models. Note that here we cannot evaluate MSP as the network was never trained to produce the predictive probabilities. As shown, we first observe that, even without task-specific fine-tuning, the AUROC scores are either very close or better to Table 1, indicating that pre-trained models work well for OOD detection out of the box. Secondly, we observe that RMD outperforms MD for all benchmarks with different pre-trained models with margins varying between 3.17 points to 16.5 points. For the CIFAR-100 vs CIFAR-10 benchmark BiT models provide the best performance followed by CLIP and Vision Transformer. BiT with RMD achieves

<sup>1</sup>We observed that the AUROC for MD changes a lot during training of the 1D CNN genomics model. We report the performance based on the model checkpoint at the end of the training without any hyperparameter tuning using validation set. See Section 4.3 for details.

<sup>2</sup>The BERT model used for the genomics benchmark is pre-trained on the genomics data with the standard masked language modeling method.

significantly higher AUROC (84.60%) in comparison to the Wide ResNet baseline model (81.01%). For CIFAR-10 vs CIFAR-100, using pre-trained CLIP, RMD achieves 91.19% AUROC, higher than any of the other methods considered. Finally, it is worth noting that the gains provided by RMD are very prominent for genomics and CLINC intent benchmark when using BERT pre-trained features.

Benchmark	MD	RMD
ViT-B <sub>16</sub> Pre-trained		
CIFAR-100 vs CIFAR-10	67.19%	<b>79.91%</b>
CIFAR-10 vs CIFAR-100	84.88%	<b>89.73%</b>
BiT R50x1 Pre-trained		
CIFAR-100 vs CIFAR-10	81.37%	<b>84.60%</b>
CIFAR-10 vs CIFAR-100	86.70%	<b>89.87%</b>
CLIP Pre-trained		
CIFAR-100 vs CIFAR-10	71.40%	<b>81.83%</b>
CIFAR-10 vs CIFAR-100	83.57%	<b>91.19%</b>
BERT Pretrained		
Genomics OOD	48.46%	<b>60.36%</b>
CLINC Intent OOD	75.48%	<b>91.98%</b>

Table 2: Comparison of OOD-AUROC for the 4 near OOD benchmarks based on feature maps from pre-trained models. No fine-tuning involved.

**Pre-trained models with fine-tuning** We now explicitly fine-tune the pre-trained model on the in-distribution dataset optimizing for classification accuracy. Using the fine-tuned models for different benchmark, we report the performance in Table 3, comparing RMD with MD and MSP baselines. We see that the performance of the MD improves significantly after the model fine-tuning (comparing Tables 2 and 3), suggesting a deletion of disruptive non-discriminative features which existed in the pre-trained models. MD achieves close or competitive AUROC when compared to RMD for most of the task evaluated, with the notable exception of genomics OOD (see Section 4.3). In light of discussion in Section 3, we conjecture that after task-specific fine-tuning using labeled data, most of the features become discriminative between IND and OOD. It is also possible that the pre-training and finetuning regimes end up at better local minima, and that the resulting features are capable of modelling the foreground and background implicitly (without our explicit normalization using RMD). Therefore the effectiveness of RMD in such cases is limited.

### 4.3 Relative Mahalanobis is more robust

In the genomics experiments, we noticed that the OOD performance of MD is quite unstable during training of the 1D

<sup>3</sup>We observed that the AUROC for MD changes a lot during fine-tuning. We report the performance based on the model checkpoint at the end of the training. See Section 4.3 for details.

Benchmark	MD	RMD	MSP
ViT-B <sub>16</sub> Fine-tuned			
CIFAR-100 vs CIFAR-10	<b>94.42%</b>	93.09%	92.30%
CIFAR-10 vs CIFAR-100	<b>99.87%</b>	<b>98.82%</b>	<b>99.50%</b>
BiT-M R50x1 Fine-tuned			
CIFAR-100 vs CIFAR-10	81.37%	<b>84.60%</b>	81.04%
CIFAR-10 vs CIFAR-100	<b>94.57%</b>	<b>94.94%</b>	85.65%
BERT Fine-tuned			
Genomics OOD	55.87% <sup>3</sup>	<b>72.04%</b>	<b>72.02%</b>
CLINC Intent OOD	<b>97.92%</b>	<b>97.62%</b>	96.99%

Table 3: AUROC for the 4 near OOD benchmarks based on feature maps from the fine-tuned models.

CNN model and the fine-tuning of the BERT pre-trained model. The AUROC of MD increases at first during the early stages of training, followed by a decrease at later stages. Figure 2 shows the change of AUROCs for MD and RMD during the training of the 1D CNN model. The AUROC of MD quickly increases to 66.19% at step 50k, when the model is not well trained yet, with training and test accuracies being 88.59% and 82.20% respectively. As the model trains further and achieves higher training accuracy of 99.96% and higher test accuracy of 85.71% at step 500k, the AUROC for MD drops to 53.10%. On the other hand, the RMD increases as the training and test accuracies increase, and gets stabilized as the accuracy stabilizes, which is a more desirable property to have. Similarly, we observed this phenomenon in the fine-tuning of the BERT genomics model. At the early training stage, AUROC for MD achieves the peak of 77.49%, while the model is not trained well with the training and test accuracies being only 82.62% and 83.97% respectively.

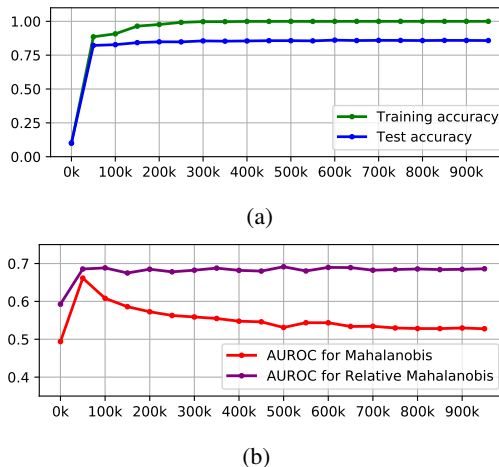


Figure 2: Comparison of MD and RMD as a function of training iterations: MD performs well in the early stage of training, but drops significantly after that, while RMD stabilizes during training, which is consistent with the pattern of in-distribution accuracy.

## Acknowledgements

We thank Zack Nado, D. Sculley and the anonymous reviewers for helpful feedback.

## References

- [1] Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.
- [2] Bishop, C. M. Novelty Detection and Neural Network Validation. *IEE Proceedings-Vision, Image and Signal processing*, 141(4):217–222, 1994.
- [3] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [4] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.
- [5] Fort, S., Ren, J., and Lakshminarayanan, B. Exploring the limits of out-of-distribution detection. *arXiv preprint arXiv:2106.03004*, 2021.
- [6] Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *ICLR*, 2017.
- [7] Hendrycks, D., Lee, K., and Mazeika, M. Using pre-training can improve model robustness and uncertainty. In *ICML*, 2019.
- [8] Hendrycks, D., Mazeika, M., and Dietterich, T. G. Deep anomaly detection with outlier exposure. *ICLR*, 2019.
- [9] Kamoi, R. and Kobayashi, K. Why is the Mahalanobis distance effective for anomaly detection? *arXiv preprint arXiv:2003.00402*, 2020.
- [10] Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., and Houlsby, N. Big transfer (BiT): General visual representation learning. *arXiv preprint arXiv:1912.11370*, 6(2):8, 2019.
- [11] Larson, S., Mahendran, A., Peper, J. J., Clarke, C., Lee, A., Hill, P., Kummerfeld, J. K., Leach, K., Laurenzani, M. A., Tang, L., et al. An evaluation dataset for intent classification and out-of-scope prediction. *arXiv preprint arXiv:1909.02027*, 2019.
- [12] Lee, K., Lee, K., Lee, H., and Shin, J. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *NeurIPS*, 2018.
- [13] Liu, J. Z., Lin, Z., Padhy, S., Tran, D., Bedrax-Weiss, T., and Lakshminarayanan, B. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *NeurIPS*, 2020.
- [14] Morningstar, W., Ham, C., Gallagher, A., Lakshminarayanan, B., Alemi, A., and Dillon, J. Density of states estimation for out of distribution detection. In *AISTATS*, 2021.
- [15] Nalisnick, E., Matsukawa, A., Teh, Y. W., and Lakshminarayanan, B. Detecting out-of-distribution inputs to deep generative models using typicality. *arXiv preprint arXiv:1906.02994*, 2019.
- [16] Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. *arXiv preprint arXiv:1705.07057*, 2017.
- [17] Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *JMLR*, 2021.
- [18] Paul, S. and Chen, P.-Y. Vision transformers are robust learners. *arXiv preprint arXiv:2105.07581*, 2021.
- [19] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.
- [20] Ren, J., Liu, P. J., Fertig, E., Snoek, J., Poplin, R., Deprieto, M. A., Dillon, J. V., and Lakshminarayanan, B. Likelihood ratios for out-of-distribution detection. *NeurIPS*, 2019.
- [21] Winkens, J., Bunel, R., Roy, A. G., Stanforth, R., Natarajan, V., Ledsam, J. R., MacWilliams, P., Kohli, P., Karthikesalingam, A., Kohl, S., Cemgil, T., Eslami, S. M. A., and Ronneberger, O. Contrastive training for improved out-of-distribution detection. *arXiv preprint arXiv:2007.05566*, 2020.
- [22] Zhang, H., Li, A., Guo, J., and Guo, Y. Hybrid models for open set recognition. *ECCV*, 2020.



## A Pseudocode for Relative Mahalanobis distance

The pseudocode for our method is shown in Algorithm 1.

---

### Algorithm 1 Relative Mahalanobis distance

---

- 1: **Input:** In-distribution train set  $\mathcal{D}_{\text{train}}^{\text{in}} = \{(\mathbf{z}_i, y_i)\}$  with  $K$  classes, in-distribution test set  $\mathcal{D}_{\text{test}}^{\text{in}} = \{\mathbf{z}'\}$ , out-of-distribution test set  $\mathcal{D}_{\text{test}}^{\text{out}} = \{\mathbf{z}'\}$ , feature extractor  $\mathbf{z} = f(\mathbf{x})$ .
  - 2: Fit the  $K$  class conditional Gaussian  $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$  using  $\mathcal{D}_{\text{train}}^{\text{in}}$ , where  $\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i:y_i=k} \mathbf{z}_i$ , for  $k = 1, \dots, K$ , and  $\boldsymbol{\Sigma} = \frac{1}{N} \sum_{k=1}^K \sum_{i:y_i=k} (\mathbf{z}_i - \boldsymbol{\mu}_k)(\mathbf{z}_i - \boldsymbol{\mu}_k)^T$ .
  - 3: Fit the background Gaussian  $\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$  using  $\mathcal{D}_{\text{train}}^{\text{in}}$  ignoring the class labels, where  $\boldsymbol{\mu}_0 = \frac{1}{N} \sum_{i=1}^N \mathbf{z}_i$  and  $\boldsymbol{\Sigma}_0 = \frac{1}{N} \sum_{i=1}^N (\mathbf{z}_i - \boldsymbol{\mu}_0)(\mathbf{z}_i - \boldsymbol{\mu}_0)^T$ .
  - 4: Compute  $\text{MD}_k(\mathbf{z}')$  based on  $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$ , for each  $\mathbf{z}' \in \mathcal{D}_{\text{test}}^{\text{in}}$  and  $\mathcal{D}_{\text{test}}^{\text{out}}$  using Eq. 2.
  - 5: Compute  $\text{MD}_0(\mathbf{z}')$  based on  $\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ , for each  $\mathbf{z}' \in \mathcal{D}_{\text{test}}^{\text{in}}$  and  $\mathcal{D}_{\text{test}}^{\text{out}}$ .
  - 6: Compute RMD confidence score  $-\min_k \{\text{MD}_k(\mathbf{z}') - \text{MD}_0(\mathbf{z}')\}$  for each  $\mathbf{z}' \in \mathcal{D}_{\text{test}}^{\text{in}}$  and  $\mathcal{D}_{\text{test}}^{\text{out}}$ .
  - 7: Compute AUROC between  $\mathcal{D}_{\text{test}}^{\text{in}}$  and  $\mathcal{D}_{\text{test}}^{\text{out}}$  based on their RMD scores.
- 

## B Additional Experimental Details

For CIFAR-10/100 experiments, we first train a Wide ResNet 28-10 model<sup>4</sup> from scratch using the in-distribution data. Next we use the publicly available pre-trained models ViT-B.16<sup>5</sup>, BiT R50x1<sup>6</sup>, and CLIP<sup>7</sup>, and replace the last layer with a classification head and fine-tune the full models using in-distribution data. We do not fine-tune CLIP model since CLIP requires paired (text, image) data for training. The fine-tuned ViT model has in-distribution test accuracy of 89.91% for CIFAR-100, and 97.48% for CIFAR-10. The fine-tuned BiT model has in-distribution test accuracy of 86.89% for CIFAR-100, and 97.66% for CIFAR-10.

For the genomics OOD benchmark, the dataset is available at Tensorflow Datasets<sup>8</sup>. The dataset contains 10 in-distribution bacteria classes, and 60 OOD classes and the input is a fixed length sequence of 250 base pairs composed by letters A, C, G and T. We first train a 1D CNN of 2000 filters of length 20 from scratch using the in-distribution data. We train the model for 1 million steps using the learning rate of  $10^{-4}$  and Adam optimizer. Next we pre-train a

<sup>4</sup><https://github.com/google/uncertainty-baselines/blob/master/baselines/cifar/deterministic.py>

<sup>5</sup>[https://github.com/google-research/vision\\_transformer](https://github.com/google-research/vision_transformer)

<sup>6</sup>[https://github.com/google-research/big\\_transfer](https://github.com/google-research/big_transfer)

<sup>7</sup><https://github.com/openai/CLIP>

<sup>8</sup>[https://www.tensorflow.org/datasets/catalog/genomics\\_ood](https://www.tensorflow.org/datasets/catalog/genomics_ood)

BERT style model by randomly masking the input token and predict the masked token using the output of the transformer encoder. The model is trained using the unlabeled training and validation data. The prediction accuracy for the masked token is 48.35%. At the fine-tuning stage, the model is fine-tuned using the in-distribution training data for 100,000 steps at the learning rate of  $10^{-4}$ , and the classification accuracy is 89.84%.

For CLINC Intent OOD, we use a standard BERT pre-trained model<sup>9</sup>, and fine-tune it using the in-distribution CLINC data for 3 epochs with the learning rate of  $10^{-4}$ . The classification accuracy is 96.53%.

## C Performance of Partial Mahalanobis distance

We compare our method with the Partial Mahalanobis distance (PMD) proposed in [9]. PMD uses a subset of eigen-bases to compute the distance score,  $\text{PMD}_S(\mathbf{z}') = \sum_{d \in S} l_d^2 / \lambda_d$ ,  $S \subset \{1, 2, \dots, D\}$ . Although  $S$  can be any subset of  $\{1, \dots, N\}$ , it was recommended to use  $S = \{1, \dots, d\}$  or  $S = \{d + 1, \dots, D\}$  corresponding to the largest or smallest Eigenvalues respectively. We compare our RMD method with the two versions of PMD using the benchmark task of CIFAR-100 vs CIFAR-10. Since there is a hyperparameter  $d$  involved in PMD, we search from  $d = 1, \dots, D$ . Figure 3a shows the AUROC when using the top eigen-bases to compute PMD. The AUROC increases as  $d$  increases and reaches to the peak of 79.72% at  $d = 76$ , and then decreases when including more dimensions. Therefore the performance of PMD method depends on the choice of  $d$ , while our method RMD is hyperparameter-free. Our method also achieves a slightly higher AUROC of 81.08% than the peak value for PMD.

We also investigate the performance of PMD when using eigen-bases corresponding to the smallest eigen-values (Figure 3b). The AUROC decreases as we exclude the top eigen-bases from the set, suggesting that the top eigen-bases are more important for the near-OOD detection. This observation supports our conjecture in Section 3 that the top eigen-bases are discriminative features and the rest are common features shared by the IND and OOD.

Another variant of the Mahalanobis distance called Marginal Mahalanobis distance (MMD) was also proposed in [9]. It fits a single Gaussian distribution to all the training data ignoring class, the same as we define the background model  $p_0$  in our RMD. Though it has a good performance for far-OOD tasks (e.g. CIFAR-10 vs SVHN) [9], it does not perform well for the near-OOD tasks, with AUROC being only 52.88% for CIFAR-100 vs CIFAR-10, and 83.81%

<sup>9</sup>[https://github.com/google/uncertainty-baselines/blob/master/baselines/clinc\\_intent/deterministic.py](https://github.com/google/uncertainty-baselines/blob/master/baselines/clinc_intent/deterministic.py)

for CIFAR-10 vs CIFAR-100.

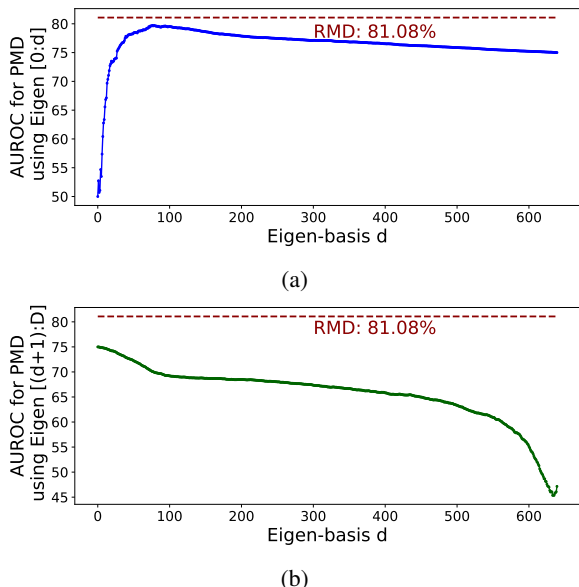


Figure 3: AUROC for Partial Mahalanobis distance (PMD) proposed in [9]. (a) PMD based on the first  $[1 : d]$  eigenbases corresponding to the  $d$  largest eigenvalues. (b) PMD based on the last  $[d : D]$  eigenbases corresponding to the smallest eigenvalues. The horizontal line indicates the AUROC for our method RMD.

## D Simulation study for the failure mode of Mahalanobis distance

We use a simple simulation to demonstrate the failure mode of Mahalanobis distance. We simulate a binary classification problem where the two classes follow a high-dimensional Gaussian distribution with different means. Specifically,  $\mathbf{x} \sim \mathcal{N}([a, 0, \dots, 0]_{1 \times D}, \sigma^2 I_{D \times D})$ , where the covariance matrix is a fixed diagonal matrix with the scalar  $\sigma$ . The mean vector has only the first dimension non-zero. To distinguish the two classes, we set  $a = -1$  for the first class,  $a = 1$  for the second class, and  $\sigma = 0.25$ . The key idea is that only the first dimension is a discriminative feature that is class-specific, whereas the remaining dimensions are non-discriminative common features that are shared by all classes. We set the number of dimensions  $D = 1024$ . To simplify the problem, we set the covariance matrix to be diagonal such that the feature dimensions are independent.

For each of the classes, we randomly sample  $n = 10,000$  data points from the given distribution for training data. For test data, we sample 100 data points from each class as the test IND data. For test OOD data, we set  $a = -3$  and  $a = +3$  and sample 100 data points from each of them. Figure 4a shows the histograms of the first dimension  $x_1$  of IND

and OOD data. The IND and OOD data points are well separated by the first dimension feature. Figure 4b shows the histogram of the remaining dimensions  $x_i, i \neq 1$ . The IND and OOD data points are not separable there, since they follow the Gaussian distribution with the same mean.

For simplicity, we first treat the  $\mathbf{x}$  as the feature map  $\mathbf{z}$ . We fit a class-conditional Gaussian  $\mathcal{N}_k$  using the training data, and compute the MD for each of the test inputs. We find that although OOD inputs in general have a greater distance than IND inputs, the two are largely overlapping. See Figure 4c for details.

The reason behind the failure mode is simple. Since the dimensions are independent, the log-likelihood of an input is the sum of the log-likelihoods of each individual dimension, i.e.  $\log p_k(\mathbf{x}) = \log p_k(x_1) + \sum_{i \neq 1} \log p_k(x_i)$ ,  $k = 1, \dots, K$ . For the discriminative feature  $x_1$ , the distributions of IND and OOD are different, so approximately  $\max_k \{\log p_k(x_1^{\text{IND}})\} > \max_k \{\log p_k(x_1^{\text{OOD}})\}$ . However, the remaining non-discriminative features  $x_i, i \neq 1$  are class-independent and both IND and OOD inputs follow the same distribution. Thus the likelihood of IND inputs based on those features will be indistinguishable from that of OOD inputs, i.e.  $\max_k \{\log p_k(x_i^{\text{IND}})\} \approx \max_k \{\log p_k(x_i^{\text{OOD}})\}, i \neq 1$ . When the number of non-discriminative features is much greater than the number of discriminative features, the log-likelihood of the former will overwhelm the latter.

Next we compute the RMD. We fit a class-independent Gaussian distribution  $\mathcal{N}_0$  using the training data regardless of the class labels, and compute the Relative Mahalanobis distance based on  $\mathcal{N}_k, k = 1, \dots, K$  (class conditional Gaussian distribution) and  $\mathcal{N}_0$  (class independent Gaussian distribution) for each of the test inputs. Using our proposed method, we are able to perfectly separate IND and OOD test inputs. See Figure 4d.

The class independent Gaussian  $\mathcal{N}_0$  helps to remove the effect of the non-discriminative features. Specifically, since those non-discriminative features are class independent, the fitted class conditional Gaussian is close to the fitted class independent Gaussian, i.e.  $\log p_k(x_i) \approx \log p_0(x_i), i \neq 1$ . Thus the two values are canceled by each other in the RMD computation, resulting in  $\max_k \{\log p_k(\mathbf{x})\} - \log p_0(\mathbf{x}) \approx \max_k \{\log p_k(x_1)\} - \log p_0(x_1)$ . For the discriminative feature, the fitted class conditional Gaussian is very different from the fitted class independent Gaussian. For IND inputs,  $\max_k \{\log p_k(x_1^{\text{IND}})\} - \log p_0(x_1^{\text{IND}}) > 0$ , since the class conditional Gaussian fits better to the IND data. For the OOD input, the difference between the two is nearly 0, since none of the two distributions fit OOD. Therefore RMD provides a better separation between IND and OOD as we have seen in Figure 4d.

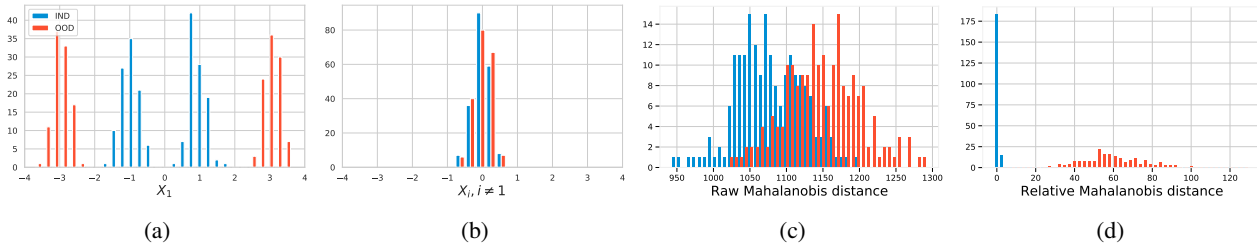


Figure 4: Simple simulation for the failure mode of Mahalanobis distance. (a) The histogram of the first dimension  $x_1$  of IND and OOD data. The two IND classes (in blue) follow  $\mathcal{N}(-1, \sigma^2)$  and  $\mathcal{N}(1, \sigma^2)$  respectively, and the two OOD classes (in red) follow  $\mathcal{N}(-3, \sigma^2)$  and  $\mathcal{N}(3, \sigma^2)$  respectively.  $\sigma = 0.25$ . (b) The histogram of  $x_i, i \neq 1$ . Both IND and OOD follow the same distribution  $\mathcal{N}(0, \sigma^2)$ . (c) The distributions of MD for IND and OOD inputs. The two distributions largely overlap. (d) The distributions of RMD for IND and OOD. The two distributions are well separated. OOD have positive values, while IND concentrate around zero.

To mimic the real scenario where the feature maps are the extracted features from the neural networks, we train simple one-layer neural networks for this binary classification task. We retrieve the feature maps of the training data, fit a class conditional Gaussian and compute MD for the test inputs. We observed the same failure mode for this case; the distributions of MD between IND and OOD largely overlap. Then we fit a class independent Gaussian and compute RMD. Using RMD, we again recover the perfect separation between the two. We expect that the intermediate layer for image, text, and genomics models also contain non-discriminative features. Therefore our proposed method is useful for overcoming this effect and improving the performance of near-OOD detection.

### E Performance on far-OOD detection

Though RMD is designed for improving near-OOD detection, we also investigate the performance of RMD on a popular far-OOD benchmark: CIFAR-10/100 vs. SVHN. Table 4 shows the results using different models, including the Wide\_ResNet model trained from scratch, a pre-trained ViT model, and a fine-tuned ViT model. For ViT pre-trained models, RMD significantly improves the AUROC (up to 27.34%); since this model has not been fine-tuned, the pre-trained model might contain background features, and we see that RMD outperforms MD in this case. For the Wide\_Resnet model trained from scratch, RMD outperforms MD and MSP on CIFAR-100 vs SVHN. For CIFAR-10 vs SVHN, MD achieves the highest AUROC of 98.16%. RMD has a comparable performance with MD with a slight lower AUROC of 95.68%. MSP has the lowest AUROC of 94.65%. For ViT fine-tuned models, MD achieves the highest AUROCs, and RMD has a slightly lower but still decent performance (93.55% for CIFAR-100 vs SVHN, and 98.70% for CIFAR-10 vs SVHN). Therefore, overall RMD does not degrade the performance over MD for far-OOD detection.

One possible reason for the slight worse performance of RMD in some cases is that, since far-OOD detection is a relatively easy task due to the different semantics and styles of OOD samples from in-distribution data, both discriminative features (like semantic features) and non-discriminative features (like background features) are useful for the detection task. Thus the role of background correction using RMD is less important for far-OOD detection task in contrast near-OOD detection task.

Benchmark	MD	RMD	MSP
Wide-Resnet 28-10			
CIFAR-100 vs SVHN	82.08%	<b>83.69%</b>	81.44%
CIFAR-10 vs SVHN	<b>98.16%</b>	95.68%	94.65%
ViT_B16 Pre-trained			
CIFAR-100 vs SVHN	61.09%	<b>88.43%</b>	NA
CIFAR-10 vs SVHN	80.77%	<b>93.60%</b>	NA
ViT_B16 Fine-tuned			
CIFAR-100 vs SVHN	<b>95.82%</b>	93.55%	93.83%
CIFAR-10 vs SVHN	<b>99.84%</b>	98.70%	99.50%

Table 4: Comparison of OOD-AUROC on the far-OOD benchmarks.