# Hiding in a Plain Sight: Out-of-Distribution Detection from Logit Space Embeddings

**Anonymous authors**
**Paper under double-blind review**

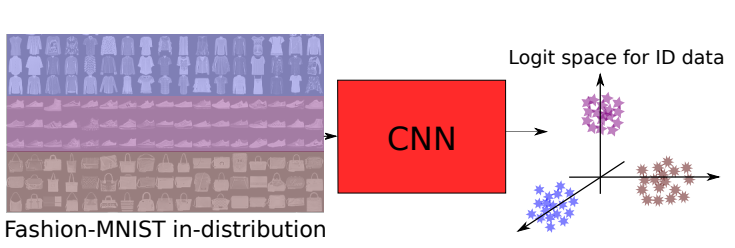
## Abstract

Although deep learning (DL) models have revolutionized the field of machine learning (ML), these classification models cannot easily distinguish the in-distribution (ID) versus the out-of-distribution (OOD) data at the test phase. This paper analyzes the landscape of ID and OOD data embeddings and demonstrates that OOD data is always embedded toward the center in the logit space. Furthermore, IDs data are embedded far from the center towards the positive regions of the logit space, thus ensuring minimal overlap between ID and OOD embeddings. Based on these observations, we propose to make the classification model sensitive to the OOD data by incorporating the configuration of the logit space into the predictive response. Hence, we estimate the distribution of the ID logits by utilizing a density estimator over the training data logits. Our proposed approach is data and architecture-agnostic and could be easily incorporated with a trained model without exposure to OOD data. We ran experiments on the popular image datasets and obtained state-of-the-art performance and an improvement of up to 10% on AUCROC on the Google genome dataset.

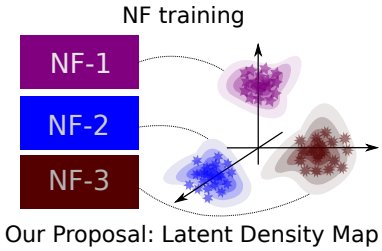
## 1 Introduction

Deep learning (DL) classification models can generalize over the discriminative features of a large amount of data, thus, providing higher classification accuracy than alternative models. The predictive response of DL models is highly accurate whenever the test data falls within the training data distribution. However, these models fail for out-of-distribution (OOD) data, as they operate under the strong assumption that the test item belongs to one of the designated classes. This incapability of DL classifiers limits their adaptation into sensitive application areas such as biomedicine. E.g., when classifying bacteria from genome sequences using a DL model, it is crucial to consider the presence of novel (i.e., OOD) bacteria. Failing to account for them may result in the incorrect classification of these novel bacteria as one of the already known types.

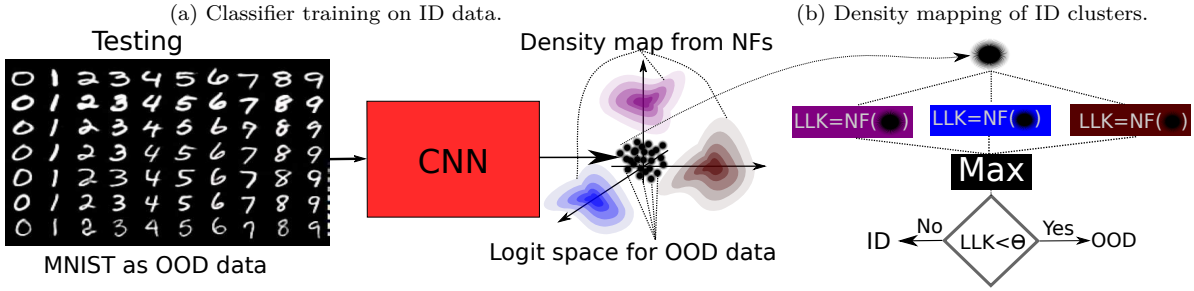
To tell OODs apart from IDs, today's deep learning (DL) architectures try to estimate the statistical uncertainty over the discriminative features of the training data However, all the previous methods implicitly assume random scattering of the OOD relative to the ID in the embedding space and fail to provide an easy-to-use solution for OOD detection. Instead, we demonstrate that a well-trained DL classifier with non-linearities that suppresses negative values (e.g., ReLU) projects the ID data into class-wise clusters toward positive regions and far from the center of the logit space (fig. 1a). Furthermore, we show (analytically and empirically) that OOD data are not arbitrarily scattered in the logit space but hidden in plain sight at its center (fig. 1c). These low-magnitude logit values for OODs result directly from their statistical independence relative to the trained model's parameter. Hence, ensuring minimal overlap between OODs and IDs. *To the best of our knowledge, this is the first work that identifies and analyzes this separation of OODs and IDs logits.* As a result of this identified separation, we can safely construct an accurate ID detector with simple architecture (cf. fig. 1c). In practice, targetable OODs as training data are necessary to depict the regions in the logit space where the OODs are projected, thereby enabling their detection. Since the distribution of OODs is unbounded, consolidating a proper training set composed solely of targetable OODs is not feasible. Thus we cast the problem as ID detection, where any non-ID is safely considered OOD. We represent the densities of each ID cluster to estimate the likelihood of any data being ID. Moreover, any data




(a) Classifier training on ID data.




(b) Density mapping of ID clusters.




(c) During the test phase, the OOD uses CNN to produce logit embeddings. If the likelihood (LLK) of these embedding is lower than a threshold ($\theta$), it is classified as OOD; otherwise, it is ID.

Figure 1: A CNN classifier model that has undergone training projects the data towards positive regions into clusters specific to each class fig. 1a. An individual density estimator is applied to each class-wise cluster formed by the logit projections fig. 1b. The identification of OOD data is achieved by analyzing the likelihood of embeddings in the logit space fig. 1c.

that attains a likelihood value below a certain threshold should fall outside any ID cluster and be considered OOD (fig. 1c).

We identified normalizing flows (NF) as a good candidate for density estimation since it provides exact likelihood without altering the dimensionality of the data Papamakarios et al. (2021). Since OODs and IDs logits are separated, the proposed method admits a simple NF architecture for each ID cluster, circumventing the need for exposure to real (or synthetic) OOD data, and does not demand alternation to the topology of the DL classifiers. The contributions of the paper are the following:
1) Analytical and empirical evidence for the ID and OOD data positioning in the logit space;
2) Novel highly effective framework for OOD detection using density estimation over the logits;
Despite having a reduced model complexity, the proposed approach (cf. fig. 1b) matches state-of-the-art (SOTA) models' performance in grayscale and colored images. Furthermore, our experiments show that it considerably improves the OOD detection performance relative to the previously reported baselines on the Google genome dataset.

## 2 Method

### 2.1 In-distribution data positioning in the logit space.

Training a deep learning (DL) classifier involves utilizing the cross-entropy loss, denoted as $H(Y,\hat{Y}) = -\sum_i Y(i)\log(\hat{Y}(i))$, to encourage the prediction ($\hat{Y}$) to closely align with the ground truth (Y). When employing one-hot encoding for both the prediction ($\hat{Y}$) and ground truth (Y), the training objective simplifies to

$$H(Y,\hat{Y}) = -\sum_i Y(i)\log(\hat{Y}(i)) = \underbrace{-Y(j)\log(\hat{Y}(j))}_{Y(j)=1, j\rightarrow\text{correct class}} - \sum_{i,i\neq j} \underbrace{Y(i)\log(\hat{Y}(i))}_{Y(i)=0, i\rightarrow\text{incorrect class}} = -\log(\hat{Y}(j)). \quad (1)$$

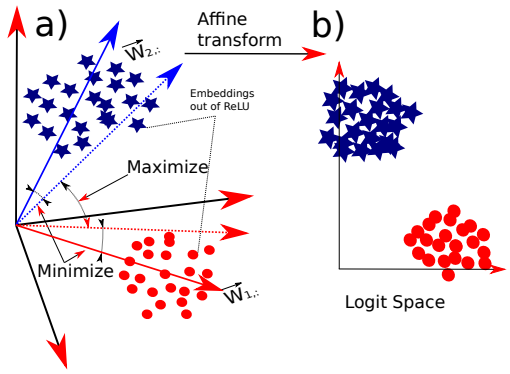



Figure 2: This toy example shows the separation of ID in a binary classification task. Figure a) contains the embeddings ($E$) rectified with a ReLU. Figure b) shows the linear separation of class-wise clustering of ID data logits ($\hat{L}$). The smaller the angle between $\vec{E}$ and $\vec{W}_{1,:}$, the higher the dot-product $\langle W_{1,i}, E_i \rangle$ Figure a); thus the more distanced from the center the ID logits are (Figure b). The bigger the angle between ($\vec{E}$) and $\vec{W}_{2,:}$, the higher the dot-product $\langle \vec{W}_{2,i}, \vec{E}_i \rangle$ (fig. 2 a), the more compact the ID logits are.

Eventually, the minimization cross-entropy loss $\big(i.e., \min[H(Y, \hat{Y})]\big)$ equivalues to maximum likelihood estimation (MLE) $\big(i.e., \min[-\log(\hat{Y}(j))]\big)$. As training progresses, the softmax layer aims to generate a response close to one for the cell corresponding to the correct class $\big(i.e., \hat{Y}(j) \to 1\big)$. Additionally, due to the property that the softmax output is confined within a simplex $\big(i.e., \hat{Y}(j)^{\uparrow} + \sum_{i, i \neq j} \hat{Y}(i)^{\downarrow} = 1\big)$, the remaining cells are pushed towards values close to zero $\big(i.e., \hat{Y}(i)_{i \neq j} \to 0\big)$.

Hence, optimization in this context can be seen as maximizing the softmax cell for the correct class while minimizing the cells for the incorrect classes. This optimization applies directly to the corresponding logit cells since softmax keeps the order of logits intact.

Specifically, the logit cell associated with the correct class aims to achieve large positive values, while the logit cells for the incorrect classes aim for small values. However, whenever ReLU is used as an activation layer, we demonstrate that the minimization process results in logit values near zero rather than small negative magnitudes.

**Theorem 1** *When training a deep learning (DL) classifier with ReLU (Rectified Linear Unit) as the non-linear activation function, the logit associated with the correct class endeavors to achieve high magnitudes of positive values $\hat{L}(j) \to +\infty$. Simultaneously, the remaining cells representing incorrect classes aim to attain low-magnitude values $\hat{L}(i)_{i \neq j} \to 0$.*

To prove this theorem, it is necessary to state the following lemma:

**Lemma 1** *In the positive region of high-dimensional space, the maximum angle two vectors can achieve is perpendicular (cf. proof in appendix A).*

To prove the restriction towards zero of the logit cells not corresponding to the correct class ($\hat{L}(i)_{i \neq j} \to 0$), it is paramount to note that the predecessor latent space ($\hat{E}(i)$) is restricted towards the positive values due to the ReLU (fig. 2.a). The layer preceding the softmax is a linear transformation of the data from high-dimensional embeddings ($\hat{E}$) to the logit space ($\hat{L} = \hat{E} \times W$, s.t: $\times$ is the matrix multiplication) with dimensions matching the number of designed classes(fig. 2.b). Since the optimizer tries to attain maximum response for the logit cell $\hat{L}[i]$, it should maximize the dot-product $\arg\max_{W[i,:]} \langle \hat{E}[:], W[i,:] \rangle$[1],s.t: $\hat{E}[:] \geq 0$.

Considering the embeddings $\hat{E}[:]$ and $W[i,:]$ as a vector in the vector space (fig. 2 a), $\langle \vec{E}[:], \vec{W}[i,:] \rangle$ maximization results in angle minimization between $\vec{E}[:]$ and $\vec{W}[i,:]$ $\big(i.e., \min \angle(\vec{W}[i,:], \vec{E}[:])\big)$ while the former always

[1] $\langle , \rangle$ indicates the dot-product

remain in the positive regions. The optimization tries to keep the direction of the vector $\vec{W}[i,:]$ similar to the cluster of vectors $\vec{E}[:]$, namely in the positive regions (fig. 2.a).

Furthermore, the optimization tries to attain a minimum response for every other logit cell $\hat{L}[j \neq i]$ that does not correspond to the correct class as $\arg\min_{W[j\neq i,:]}\langle \hat{E}[:], W[j \neq i,:]\rangle$, s.t: $\hat{E}[:] \geq 0$. Namely, maximizing the angle between $\vec{W}[j \neq i,:]$ and the cluster of vector data $\vec{E}[:]$, $\big(i.e., \max \angle(\vec{W}[j \neq i,:], \vec{E}[:])\big)$ (fig. 2.a).

Hence, the clusters belonging to different classes strive to achieve maximum angular separation from one another, and the parameter vectors $\vec{W}[i,:]$ align accordingly. As all vectors $\vec{E}[:]$ are angularly separated within the positive region, the maximum angle between these two vectors is close to perpendicularity (cf. lemma 1). Therefore, the minimized logit values $\big(\arg\min(\vec{W}[j \neq i,:], \vec{E}[:]) \approx 0\big)$ would approach assymptotically to zero during training.

Consequently, the asymptotic behavior of the data configuration in the logit space compels the data points to form compact clusters far from the center of the space, corresponding to their respective classes. This process leads to the minimization of interclass distances and the maximization of intraclass distances.

### 2.2 Out-of-distribution data positioning in the logit space

We demonstrated that optimization pushes the IDs away from the center of the logit space and toward the positive regions (cf., section 2.1). To show that OODs are not arbitrarily scattered in the logit space but projected towards the center, we show that the interaction of the parameters of the model ($\omega$) and the OOD data ($x_{OOD}$) is upper-bounded by the independence consistency (cf., definition 1) between these two random variables (r.v) $(\omega \perp x_{OOD})$[2].

**Definition 1** ***(Consistently independent)*** *Given two r.v $x, \omega \in R$ where the probability density function (pdf) of $x$ is fixed, whereas the pdf of $\omega$ is unbounded. These two r.v are systematically independent if the expectation of their covariance is zero (i.e., $E[cov(x,\omega)]^2) = 0$. Additionally, these two r.v maintain this independence consistently whenever the variability of their covariance is zero (i.e., $Var[cov(x,\omega)] = 0$). Hence two r.v are consistently independent if and only if $E[cov(x,\omega)]^2 + Var[cov(x,\omega)] = 0$.*

Projection of OODs towards the center of the logit space enables easy separation from the IDs (section 2.3). However, to establish the bounds of OOD in the logit space, we start by showing that a dot-product between two variables acts as a lower bound for their covariance value (corollary 1).

**Corollary 1** *Given two r.v $(x, \omega \in R)$, the magnitude of their dot-product is a lower bound for the magnitude of their covariance (cf. eq. (2)) (cf. proof in appendix B).*

$$\big|\langle x, \omega\rangle\big|^{3} \leq \big|cov(x,\omega)\big|, \forall x, \omega \in \mathbb{R} \tag{2}$$

Thus, the covariance between these two entities (i.e., weights ($\omega$) and training data $x_{\text{ID}}$) are maximized while their dot-product magnitude is maximized during the training process. While the $\big|\text{cov}(x,\omega)\big|$ is maximized, the $\big|\text{cov}(x_{\text{OOD}}, x_{\text{ID}})\big|$ is assumed to be minimal since $(x_{\text{OOD}} \perp x_{\text{ID}})$) the following corollary can be easily derived (corollary 2). A high covariance value magnitude characterizes two variables covary together in their respective spaces.

**Corollary 2** ***(Co-variability)*** *Since the OOD $(x_{OOD})$ and ID $(x_{ID})$ data come from two different distributions altogether, one can assume that their covariance is minimal (i.e., $cov(x_{OOD}, x_{ID}) \approx 0$). Given that IDs covary with $\omega$ but not with the OODs, we prove that OODs do not covary with the $\omega$ (i.e., $cov(x_{OOD}, \omega) \approx 0$) (cf. proof in appendix C).*

The training process makes the model parameters ($\omega$) more statistically dependent on the IDs ($x_{\text{ID}}$), while the latter are supposed to be consistently independent of the OODs ($x_{\text{OOD}}$). Since the magnitude of the

[2] $\perp$ for two r.v means statistical independence.
[3] $\langle,\rangle$ indicates the dot-product

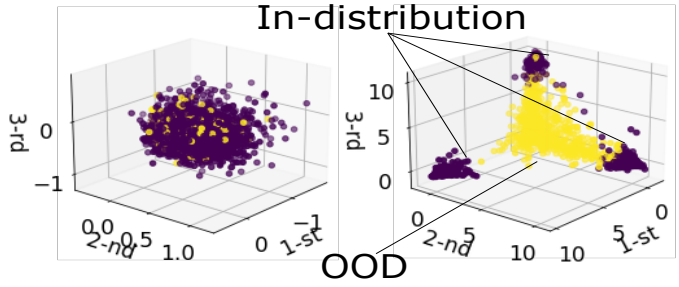



Figure 3: CIFAR-3 as ID and SVHN as OOD. *Left*: Before the training, both ID and OOD maintain the tendency towards the center of the latent space. *Right*: After the training, the ID data are clustered, whereas the OOD persists towards the center.

logits is a result of dot-product, this can be upper-bounded by the independence consistency between the OODs and $\omega$. Utilizing this low covariability of OODs ($x_{\text{OOD}}$) and the model weights ($\omega$), we derived an upper bound for the expectation of their dot-product (theorem 2). Hence, by taking the expectation on both sides of eq. (2), one can prove the following theorem (theorem 2).

**Theorem 2 (Expectations of OOD embeddings)** *Given two random variables $(x_{OOD}, \omega)$ whose covariance magnitude is at the proximity of zero, the expectation of their dot-product is upper bounded by their independence consistency (cf. proof in appendix D):*

$$\Big|E\big[\langle x_{OOD}, \omega\rangle\big]\Big| \leq \Big|\sqrt{E[cov(x_{OOD}, \omega)]^2 + Var[cov(x_{OOD}, \omega)]}\Big|, \forall x_{OOD}, \omega \in R \tag{3}$$

The r.h.s of eq. (26) indicates the OODs expected distance from the center of the logit space. The l.h.s eq. (26) is the independence consistency between OOD data and the parameters of the models. Therefore, the better the trained model is (i.e., $\big|\langle x, \omega\rangle\big|^{\uparrow}$ is high), the farther from the center the IDs are clustered (cf., section 2.1). The more different OODs are from the IDs, the lower their covariance with the IDs hence the lower $\big|\text{cov}(x_{\text{OOD}}, \omega)\big|$, leading to lower l.h.s of eq. (26). Hence, unlike ID data, the OOD data embeddings will not be able to produce high magnitude values for any of the logit cells, and their embeddings are squeezed more towards the center and well separated from IDs.

To restrict the visualization to 3D space, a three-class model (Resnet-34) is investigated on the CIFAR-3[4] vs. SVHN scenario. The untrained Resnet-34 model with three output classes whose weights are randomly initialized projects both CIFAR-3 and SVHN datasets towards the center of the logit space (cf., fig. 3).

This concentration of OODs logit the center is further validated empirically for both image and genome dataset (figs. 5 to 9 in Appendix). In fig. 4a in the appendix, we included a kernel density estimation (KDE) over the first logit cell for both OODs and IDs of class one. Before training, both datasets are unencountered to the model; therefore, their response is concentrated toward zero. After the training, most of the IDs response is pushed towards high positive values while the OODs remain intact (fig. 4b in Appendix).

### 2.3 Model for OOD detection

After establishing two distinct regions for OOD and ID data that do not overlap, the next step involves detecting and exposing OODs during the testing phase.

In practice, accurately defining the boundaries of the OOD region is challenging because OOD data is typically unavailable. However, even after training the classifier, we retain access to the embeddings of the ID training data. Leveraging this information, it becomes feasible to delineate the ID regions using density estimation techniques.

[4]Only three classes from CIFAR-10

Therefore, we address the issue of OOD absence by developing an ID detection system that acts as a one-class classifier, treating any data identified as non-ID to be OOD. To delineate the regions corresponding to ID, we utilize the density representation of each ID cluster to assess the probability of a given data point being classified as ID. By creating a density map based on the occurrence of ID embeddings from the training data, we capture the uncertainty associated with the discriminative features.

Normalizing flows (NF) offers an appropriate framework for density estimation due to their ability to estimate the likelihood for a given data while preserving the original dimensionality of the vector space Papamakarios et al. (2021). While alternative parametric density estimators like the Gaussian distribution provide similar capabilities, they rely on a more restrictive assumption that the data distribution follows an elliptic shape. In contrast, NF does not impose any prior assumptions about the shape of the target distribution. Instead, they employ a bijective transformation, denoted as $u = T_\theta^{-1}(x)$, to map a simple base distribution $P_U(u)$ which could be Gaussian to match the desired target distribution $P_X(x)$ (cf. eq. (4)).

$$P_X(x) = P_U(T_\theta^{-1}(x))\left|det\{J_{T_\theta^{-1}}(T_\theta^{-1}(x))\}\right| \tag{4}$$

The density estimation augments multiple runs of individual data on the classifier model as NF extrapolates the density values in the continuity of the latent space. NF can interpolate a likelihood value from positioning a single data in the latent space, reflecting the frequency of past occurrences.

The empirical distribution of the logits is already known to have the same modality as the number of clusters. Under this domain knowledge, it is more beneficiary to employ multiple NF for each class-wise ID cluster with simple architecture instead of a single but complex NF (cf., algorithm 2). Utilizing NF for each ID cluster (cf., algorithm 2 and fig. 10), it is possible to maintain a high likelihood at high-density regions and, by default, a low likelihood elsewhere, including the OOD region (cf., algorithm 2 and appendix I). Then any data whose likelihood is below a certain threshold are considered OOD (cf., fig. 1 and algorithm 1).

**Algorithm 1** OOD detection

**Input:** Trained classifier that produces logits $\hat{L}(x) = F_\theta(x)$. Individual data $x$. The number of classes K. K different trained $NF_{[1\cdot K]}$ models. Threshold value $\theta$.

**procedure** D(x)
  $\hat{L}(x) \leftarrow F_\theta(x)$ ▷ Get the embeddings in the logit space
  $LogLikehood_{Max} \leftarrow 0$ ▷ Get the embeddings in the logit space
  **for** each $c = 1, 2, \ldots K$ **do** ▷ Iterate each $NF_{[1\cdot K]}$
    $\hat{\text{LLK}} \leftarrow NF_{[c]}(L(x))$ ▷ Get the likelihood for class c
    **if** $L\hat{L}K = LogLikehood_{Max}$ **then** ▷ If the prediction matches the given label
      $LogLikehood_{Max} = L\hat{L}K$ ▷ Train the NF over the ID cluster of class $c$
  **if** $LogLikehood_{Max} \geq \theta$ **then** ▷ Check if $x$ is an ID
    $x \rightarrow ID$
  **else**
    $x \rightarrow OOD$ ▷ If the data is not ID, then it is OOD

## 3 Related works

**Theoretical studies:** In a recent exploration of the learnability of the OOD task using the lens of probably approximately correct (PAC) theory, researchers conducted an insightful analysis Fang et al. (2022). Furthermore, a recent empirical investigation focused on the transferability of ID training to OOD detection Wenzel et al. (2022). An important finding from this research was the asserted correlation between enhanced ID training and improved OOD detection performance. Additionally, examination of the OOD region within the softmax space has been explored in related works Pearce et al. (2021); Frosst et al. (2019).

**Classification-based:** Detecting OOD samples using a classifier trained on ID data relies on prediction scores that are used to tell the ID classes apart. Early works use probability values from softmax as a

**Algorithm 2** Training of NF for each ID cluster

**Input:** Trained classifier that produces logits $\hat{L}(x) = F_\theta(x)$. ID annotated training data $[c, x] \in X$. The number of classes K. K different $NF_{[1..K]}$ models.

**procedure** TRAIN INDIVIDAL NF
  **for** each round $t = 1, 2, \ldots$ **do**
    **for** each data $c, x \in X$ **do** ▷ Iterate the dataset in batches
      $\hat{L}(x) \leftarrow F_\theta(x)$ ▷ Get the embeddings in the logit space
      $\hat{c} \leftarrow \arg\max L(x)$ ▷ Get the class prediction
      **if** $\hat{c} = c$ **then** ▷ If the prediction matches the given label
        Train $\rightarrow NF_{[c]}[L(x)]$ ▷ Train the c-th NF over the ID cluster of class $c$

common choice. The baseline approach for OOD detection involved using the maximum softmax output as a guiding principle . Other methods have been explored to estimate uncertainty in predictive responses by creating an ensemble of models Vyas et al. (2018). Nevertheless, this ensemble-based approach requires training and storing multiple models.

Alternatively, some methods aim to estimate data and model uncertainty using an ensemble of models. The generalized uncertainty is then distilled into a single model, and OOD samples are detected using the distilled uncertainty from this single model instead of an ensemble Vadera et al. (2020a) Malinin et al. (2019); Vadera et al. (2020b); Depeweg et al. (2017); Lakshminarayanan et al. (2016).

ODIN is another method that enhances sensitivity towards OOD data by maximizing the entropy of softmax responses Liang et al. (2020). ODIN increases OOD detection capabilities by combining a calibrated softmax with input perturbation. However, ODIN requires exposure to OOD data for training.

Another method based on softmax output proposes a regret score, calculated as the logarithm of the sum of fine-tuned probability values obtained from softmax (p(y|x)). However, softmax itself cannot capture sufficient uncertainty, making its application in OOD detection sub-optimal Gal & Ghahramani (2016); Hendrycks & Gimpel (2016); Liu et al. (2020); Sun et al. (2021); Hendrycks et al. (2019); Sastry & Oore (2020); Yu & Aizawa (2019); Hein et al. (2019).

Another model utilizes the Mahalanobis distance (MD) between the test data and the per-class center in the latent space Lee et al. (2018). While this method performs well on popular image datasets, it can fail on complex datasets where the logits do not follow a Gaussian distribution. Nonlinear boundaries in the embedding space must be considered, as the MD accounts only for isocontours of elliptic shape.

Energy-based models have also shown potential for OOD detection Liu et al. (2020), although training them can be challenging. Another approach involves using Gram matrices of different orders from each layer's output Sastry & Oore (2020). However, its performance depends on the specific order chosen for the Gram matrix of each layer.

**Generative models:** Deep generative models have been extensively studied for out-of-distribution (OOD) detection due to their ability to represent high-dimensional data uncertainty in a parametric form Serrà et al. (2020); Xiao et al. (2020); Wang et al. (2020); Choi et al. (2019); Kim et al. (2021); Schirrmeister et al. (2020); Abati et al. (2019); Ren et al. (2019); Nalisnick et al. (2018). An additional advantage of these models is their ability to operate without relying on labels, which can often be challenging to obtain.

However, these models do not generalize over the discriminative features since their training process does not involve the context of the training data. Consequently, when trained on image data, these models tend to generalize based on pixel correlation values alone, without considering discriminative features. Hence, these models may assign likelihood values to OOD data that are similar to or even higher than those assigned to ID data Kirichenko et al. (2020); Nalisnick et al. (2018); Kim et al. (2021); Xiao et al. (2020); Liu et al. (2020); Choi et al. (2019); Ren et al. (2019); Schirrmeister et al. (2020); Wang et al. (2020); Hendrycks et al. (2019); Serrà et al. (2020); Hsu et al. (2020); Sun et al. (2021); Sastry & Oore (2020).

Instead, some recent work tries to leverage contrastive learning for feature distillation and then train a density estimator Liu & Abbeel (2020).

An improved alternative model estimates the ratio of the training data likelihood over the likelihood of their noisy version Ren et al. (2019). The authors argue that performing density estimation over a noisy dataset equals uncertainty estimation over non-discriminative features. By incorporating uncertainty in the denominator, this method aims to reduce the likelihood assigned to background noise, thereby amplifying the core features. The effectiveness of this approach relies on the level of noise introduced during the training data generation process. Consequently, determining the appropriate noise level requires exposure to OOD data through simulation or real-world acquisition.

In contrast to previous approaches, our proposed solution anticipates the arrangement of both in-distribution (ID) and out-of-distribution (OOD) data within the logit space. This eliminates the necessity for complex models to achieve high performance in OOD detection.

## 4 Experiments

In order to evaluate the effectiveness of the suggested approach, the ID and OOD datasets must exhibit a significant degree of similarity while being semantically different. We performed experiments using diverse image datasets, encompassing grayscale images, colored images, and a genome dataset.

The DL classifier model is trained to classify the ID training data correctly and, by extension, to obtain their logit space representation (cf. Appendix appendices F to H). All the non-linearities in our method are ReLU to ensure maximum displacement from the center of the logit space of ID embeddings. In the case of the grayscale images, a small DL model with three convolutional layers and two fully connected layers (cf. table 6 and appendix F.1) is utilized. While in the remaining experiments, a Resnet-34 He et al. (2015) is trained as a DL classifier model (cf. appendices G and H).

Thereafter, the softmax layer of the classifier model was removed, and the remaining parameters were frozen, meaning their gradients were set to zero. The trained classifier then operates as a mapping function to convert the training data into logits. The logits of the ID training data are used to train individual NF with identical architectures. Each NF is dedicated to one of the classes (cf. fig. 1b, algorithm 2, and appendix I). Real-valued non-volume preserving (RealNVP) is the NF model choice Dinh et al. (2016), which consists of multiple MLP layers (cf. table 12). The base distribution (*i.e.*, $P_X(x)$) for the NF is a multivariate standard Gaussian with the masking as in I.

The performance of the method in detecting out-of-distribution (OOD) data was evaluated using the receive operating characteristic (ROC) and precision-recall curve (PRC) (cf. table 1) as well as true negative rate (TNR) at 95% true positive rate (TPR) (cf. table 2).

Table 1: Performance comparison on the Fashion-MNIST (ID) vs. MNIST (OOD) on the first part and Genome dataset on the second. The mean performance is reported together with the variance of ten rounds in the brackets. Apart from our method result, the rest of the results are from the Ren et al. (2019)

| **Dataset** $\rightarrow$ | Fashion-MNIST vs MNIST | | Google Genome | |
|---|---|---|---|---|
| **Methods** $\downarrow$ | AUCROC (%)$\uparrow$ | AUCPRC (%)$\uparrow$ | AUCROC (%)$\uparrow$ | AUCPRC (%)$\uparrow$ |
| **Our Method (Ours)** | **99.2 (0.1)** | **99.4 (0.1)** | **84.1 (0.9)** | **85.9 (0.8)** |
| Likelihood Ratio ($\mu$) | 97.3 (3.1) | 95.1 (6.3) | 73.2 (1.5) | 71.9 (1.7) |
| Likelihood Ratio ($\mu, \lambda$) | **99.4 (0.1)** | **99.3 (0.2**) | 75.5 (0.1) | 71.9 (0.6) |
| Mahalanobis distance | 94.2 (1.7) | 92.8 (2.1) | 52.5 (1.0) | 50.3 (0.7) |
| $p(\hat{y}\|x)$ with calibration | 90.4 (2.3) | 89.5 (2.3) | 66.9 (0.5) | 63.5 (0.4) |
| ODIN | 75.2 (6.9) | 76.3 (6.2) | 69.7 (1) | 67.1 (1.2) |
| Ensemble, 5 classifiers | 83.9 (1.0) | 83.3 (0.9) | 68.2 (0.2) | 64.7 (0.2) |
| Ensemble, 10 classifiers | 85.1 (0.7) | 84.4 (0.6) | 69 (0.1) | 65.5 (0.2) |
| Ensemble, 20 classifiers | 85.7 (0.5) | 84.9 (0.4)) | 69.5 (0.1) | 65.9 (0.1) |

Table 2: Performance comparison on the CIFAR-10 and SVHN.

| Baseline/Odin/Mahalanobis/Gram/Ours | | | |
|---|---|---|---|
| ID (model) | OOD | TNR at TPR 95% (%)↑ | AUCROC (%)↑ |
| CIFAR-10 (ResNet) | iSUN | 44.6 / 73.2 / 97.8 / 99.3 / **100** | 91.0 / 94.0 / 99.5 / 99.8 / **100** |
| | LSUN(C) | 48.6 / 62.0 / 81.3 / 89.8 / **99.9** | 91.9 / 91.2 / 96.7 / 97.8 / **99.9** |
| | TinyImgNet(C) | 46.4 / 68.7 / 92.0 / 96.7 / **99.0** | 91.4 / 93.1 / 98.6 / 99.2 / **99.8** |
| | SVHN | 50.5 / 70.3 / 87.8 / 97.6 / **98.0** | 89.9 / 96.7 / 99.1 / 99.5 / **99.8** |
| | CIFAR-100 | 33.3 / 42.0 / 41.6 / 32.9 / **64.8** | 86.4 / 85.8 / 88.2 / 79.0 / **95.0** |
| SVHN(ResNet) | iSUN | 77.1 / 79.1 / 99.7 / 99.4 / **100** | 92.2 / 91.4 / 99.8 / 99.8 / **100** |
| | LSUN(R) | 74.3 / 77.3 / 99.9 / 99.6 / **100** | 91.6 / 89.4 / 99.9 / 99.8 / **100** |
| | TinyImgNet(R) | 79.0 / 82.0 / **99.9** / 99.3 / 97.5 | 93.5 / 92.0 / **99.9** / 99.7 / 99.5 |
| | CIFAR-10 | 78.3 / 79.8 / **98.4** / 85.8 / **98.0** | 92.9 / 92.1 / 99.3 / 97.3 / **99.4** |

### 4.1 Ablation study

The effectiveness of the proposed approach is primarily attributed to the capability of the classifier to push the ID logits far from the center, and the capacity of the NF technique accurately maps the density of the ID embeddings.

The impact of the classifier on OOD detection performance can be assessed by considering the flexibility of the classifier. One critical contributory factor to the flexibility of the classifier is the number of parameters and the architecture.

When conducting the genome experiment, an increase in complexity of the classifier model, characterized by a higher number of parameters, leads to a noticeable improvement in OOD detection performance (table 3). While the improvement in validation accuracy for the classifier on ID data may not be significant as the classifier complexity increases, the embeddings produced by these classifiers exhibit a progressive separation from OOD examples (cf. table 3).

Table 3: Performance of OOD detection over the genome dataset over an increasingly more flexible classifier (increasing the number of parameters for Resnet-34).

| **Nr Parameters (M)** → | 3.7 | 3.9 | 5.3 | 5.5 |
|---|---|---|---|---|
| AUCROC (%)↑ | 75.9 | 75.8 | 82.1 | 86 |
| AUCPRC (%)↑ | 76.6 | 77.7 | 80.6 | 87.9 |

In order to assess the significance of NF performance as a density estimator, its architecture and the amount of training data are investigated. The success of the RealNVP architecture, which is the preferred choice for the NF model, is attributed to its affine coupling technique.

RealNVP Dinh et al. (2016) employs a masking operation to apply a separate nonlinear transformation to each dimension of the base distribution (i.e., in $P_X(x)$ eqs. (4) and (5)).

$$y = m_i \odot x + (1 - m_i)\{x \odot exp(NN_t(m_i \odot x)) + NN_s(m_i \odot x)\} \tag{5}$$

By using two distinct deep learning models ($NN_t$ and $NN_t$ in eq. (5)), RealNVP transforms the multi-dimensional base distribution (i.e., $P_X(x)$ in eq. (4)) into the target distribution. This transformation is performed in multiple stages, where only one dimension is transformed at each stage while the others remain unchanged.

An ablation is conducted to examine how different coupling of based distribution dimensions impact the OOD detection performance. Dimension coupling involves jointly transforming the coupled dimensions of the based distribution (i.e., $P_X(x)$) instead of treating them independently. While maintaining the same model architecture for each dimension of the NF, three different masking methods (figs. 11 to 13) that enable different types of coupling for the base distribution dimensions are tested (cf. table 4). In addition

Table 4: OOD detection performance fashion-MNIST (ID) vs. MNIST (OOD) using different masks and repetition for each mask (cf. figs. 11 to 13). The first column indicates the type of mask utilized. The second column indicates the metric and the rest indicates the repetition number.

| **Mask type and repetion.** | | | | | |
|---|---|---|---|---|---|
| **Mask type** $\downarrow$ | **Repetition** $\rightarrow$ | 1 | 2 | 3 | 4 |
| Mask 1 | AUCROC $^{(\%)\uparrow}$ | 96.2 | 99.5 | 99.3 | 98.7 |
| | AUCPRC$^{(\%)\uparrow}$ | 97.0 | 99.5 | 99.5 | 99.1 |
| Mask 2 | AUCROC $^{(\%)\uparrow}$ | 98.8 | 99.2 | 99.2 | 99.2 |
| | AUCPRC$^{(\%)\uparrow}$ | 98.5 | 99.4 | 99.4 | 99.3 |
| Mask 3 | AUCROC $^{(\%)\uparrow}$ | 99.0 | 99.3 | 99.2 | 99.1 |
| | AUCPRC$^{(\%)\uparrow}$ | 99.3 | 99.5 | 99.4 | 99.3 |

Table 5: The first column indicates the percentage of the ID data fashion-MNIST (ID) utilized for the training of the NF. The second and third columns indicate the model's performance at OOD (MNIST data) detection.

| **Data size on training NF.** | | |
|---|---|---|
| **Data ratio** | AUCPRC$^{(\%)\uparrow}$ | AUCPRC$^{(\%)\uparrow}$ |
| 0.1% | 71 | 75.5 |
| 1% | 92.9 | 94.2 |
| 10% | 98.2 | 98.7 |
| 25% | 98.5 | 98.9 |
| 50% | 99.1 | 99.4 |

to masking, the number of repetitions of each MLP unit (cf. eq. (40)) is another crucial component of the NF. Hence, different masking operations are tested for up to four repetitions to investigate (cf. table 4).

Understanding the impact of training data size remains of highly practical relevance. Therefore instead of training the NFs with all the training data utilized to train the classifier, a progressively smaller part is tried. The OOD detection performance is evaluated on the entire test data set (cf. table 5).

## 5 Discussion and Conclusion

Our work addresses OOD detection by building an accurate ID detection from logits as a one-class classifier and considering any non-ID as an OOD. This detection is possible as the method considers a maximal separation of OODs and IDs in the logit space. Under the assumption that their discriminative features originate from an altogether different distribution relative to the training data, it is possible to anticipate the interaction of OODs with the trained classifier parameters (section 2.2).

Whenever negative values are suppressed through the nonlinear function (i.e., ReLU), ID data are maximally distanced from the center towards the positive regions of the logit space (section 2.1). We show, that the better the performance of the classifier on the ID data, the more distance their embeddings have from the center of the logit space, and the more compact their class-wise clusters are (figs. 5 to 8 in Appendix). Another critically important aspect of an accurate classifier is that it embeds the OODs towards the center of the logit space such as they do not overlap with the IDs (section 2.2). The detection performance of OOD is driven by the scale of their statistical independence relative to the IDs (low covariability) and the accuracy of the classifier (high covariability between parameters and IDs). The more consistent this independence between OODs and IDs, the lower the expected magnitude of the OOD logits (eq. (26)). Combining these two key drivers enables decoupling the IDs from the OODs in the logit space.

The proposed method demonstrates SOTA performance on FPR at 95% TPR, AUCPRC, and AUCROC on both images and genome datasets. The primary factors driving the success of the proposed method are twofold: the accuracy of the classifier in the ID data and the thorough density mapping of the ID logit

embeddings. Since ID data are sufficient for training classifiers and NF, the proposed method does not require exposure to simulated or gathered data intended to be OOD as in Tack et al. (2020); Winkens et al. (2020); Hendrycks et al. (2019). Using logit embeddings enables the formation of compact class-wise clusters. Utilizing a dedicated NF for each cluster in the logit space allows effective density mapping. By employing individual NFs for each designated class, the complexity of the architecture for each NF can be reduced significantly compared to using a single NF for all clusters combined.

This work has identified the importance of using ReLU as an activation function to separate OODs from IDs in the logit space and clustering the IDs compactedly towards the positive regions. However, to ensure an even further separation between OODs and IDs, a more thorough analysis of transcendental functions that suppress negative values beyond ReLU is required.

Another promising future avenue is to leverage and replace the softmax with the NF setup (cf. 1) for a better uncertainty estimation within the conformal prediction setup Angelopoulos & Bates (2022). Given that the proposed configuration of NF (cf. 1) incorporates more effectively data uncertainty into the prediction response than softmax, one can use these NF scores to better calibrate the prediction via conformal prediction Angelopoulos & Bates (2022).

## References

Davide Abati, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Latent space autoregression for novelty detection, 2019.

Anastasios N. Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification, 2022.

Hyunsun Choi, Eric Jang, and Alexander A. Alemi. Waic, but why? generative ensembles for robust anomaly detection, 2019.

Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning, 2017.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. *CoRR*, abs/1605.08803, 2016.

Zhen Fang, Yixuan Li, Jie Lu, Jiahua Dong, Bo Han, and Feng Liu. Is out-of-distribution detection learnable? 2022.

Nicholas Frosst, Nicolas Papernot, and Geoffrey Hinton. Analyzing and improving representations with the soft nearest neighbor loss, 2019.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *PMLR*, pp. 1050–1059, New York, New York, USA, 06 2016. PMLR.

Ian J. Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay Shet. Multi-digit number recognition from street view imagery using deep convolutional neural networks. 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem, 2019.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. 2016.

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

Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data, 2020.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018.

Keunseo Kim, JunCheol Shin, and Heeyoung Kim. Locally most powerful bayesian test for out-of-distribution detection using deep generative models. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 14913–14924. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/7d3e28d14440d6c07f73b7557e3d9602-Paper.pdf.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Why normalizing flows fail to detect out-of-distribution data, 2020.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2016.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

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

Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks, 2020.

Hao Liu and Pieter Abbeel. Hybrid discriminative-generative training via contrastive learning, 2020.

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

Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. Ensemble distribution distillation, 2019.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know?, 2018.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference, 2021.

Tim Pearce, Alexandra Brintrup, and Jun Zhu. Understanding softmax confidence and uncertainty, 2021.

Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A. DePristo, Joshua V. Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection, 2019.

Chandramouli Shama Sastry and Sageev Oore. Detecting out-of-distribution examples with in-distribution examples and gram matrices, 2020.

Robin Tibor Schirrmeister, Yuxuan Zhou, Tonio Ball, and Dan Zhang. Understanding anomaly detection with deep invertible networks through hierarchies of distributions and features, 2020.

Joan Serrà, David Álvarez, Vicenç Gómez, Olga Slizovskaia, José F. Núñez, and Jordi Luque. Input complexity and out-of-distribution detection with likelihood-based generative models, 2020.

Yiyou Sun, Chuan Guo, and Yixuan Li. React: Out-of-distribution detection with rectified activations, 2021.

Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances, 2020.

Meet P. Vadera, Adam D. Cobb, Brian Jalaian, and Benjamin M. Marlin. Ursabench: Comprehensive benchmarking of approximate bayesian inference methods for deep neural networks, 2020a.

Meet P. Vadera, Brian Jalaian, and Benjamin M. Marlin. Generalized bayesian posterior expectation distillation for deep neural networks, 2020b.

Joost van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network, 2020.

Apoorv Vyas, Nataraj Jammalamadaka, Xia Zhu, Dipankar Das, Bharat Kaul, and Theodore L. Willke. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers, 2018.

Ziyu Wang, Bin Dai, David Wipf, and Jun Zhu. Further analysis of outlier detection with deep generative models, 2020.

Florian Wenzel, Andrea Dittadi, Peter Vincent Gehler, Carl-Johann Simon-Gabriel, Max Horn, Dominik Zietlow, David Kernert, Chris Russell, Thomas Brox, Bernt Schiele, Bernhard Schölkopf, and Francesco Locatello. Assaying out-of-distribution generalization in transfer learning, 2022.

Jim Winkens, Rudy Bunel, Abhijit Guha Roy, Robert Stanforth, Vivek Natarajan, Joseph R. Ledsam, Patricia MacWilliams, Pushmeet Kohli, Alan Karthikesalingam, Simon Kohl, Taylan Cemgil, S. M. Ali Eslami, and Olaf Ronneberger. Contrastive training for improved out-of-distribution detection, 2020.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. 2017. URL `http://arxiv.org/abs/1708.07747`. cite arxiv:1708.07747Comment: Dataset is freely available at https://github.com/zalandoresearch/fashion-mnist Benchmark is available at http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/.

Zhisheng Xiao, Qing Yan, and Yali Amit. Likelihood regret: An out-of-distribution detection score for variational auto-encoder, 2020.

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