# Typicality-based point OOD detection with contrastive learning

Nawid Keshtmand*[1], Raul Santos-Rodriguez[1], and Jonathan Lawry[1]

[1]University of Bristol, UK

## Abstract

Typicality-based inference methods for OOD detection find a typical value (often the mean value) of a model statistic from the training data and then flag test points as anomalous if the model statistic of the test data point deviates significantly from the typical value. These methods are effective for detecting a group of OOD data points when OOD data points are labeled into groups, but ineffective for the detection of individual OOD data points. In this paper, we extend typicality-based inference to be effective for point OOD detection by first utilizing latent features learned with contrastive learning and then leveraging the nearest neighbors of a test data point to provide additional context used for point OOD detection. Additionally, we use a one-dimensional variant of the Mahalanobis Distance as our statistic which has not previously been used in a typicality-based inference method. This typicality-based inference approach is shown to improve point OOD detection relative to several benchmarks.

## 1 Introduction

Neural networks have been shown to achieve high accuracy on a variety of different tasks, however, they are still prone to error when faced with data drawn from a distribution different from that of the training distribution [1]. This poses a considerable obstacle to the real-world deployment of neural networks in safety-critical applications where an incorrect prediction can have extreme consequences [2]. Hence, it is important to be able to distinguish in-distribution (ID) data, from which the training data was drawn, from out-of-distribution (OOD) data. OOD detection, outlier detection, or anomaly detection [2] can be categorized into two types; the detection of individual OOD data points (point OOD detection) and a group of OOD data points (group OOD detection) where all the data points in the group are considered OOD.

A popular approach to point OOD detection is to learn latent features of the data (usually by training a neural network model) and then use an OOD inference method on the latent features of test data points. Self-supervised contrastive learning is an effective way to learn the latent features of the data [3]. Contrastive learning achieves this by training a feature extractor to discriminate between different individual instances of data. Using a data augmentation specific to the particular data modality, contrastive learning involves initially creating two versions of the same data instance, commonly referred to as positives. A model trained with contrastive learning is then optimized to move each instance closer to its positive, and further away from other negative instances of the data [4].

**Related work** Contrastive learning approaches to OOD detection involve training a model with a variant of contrastive learning such as the unsupervised instance discrimination or the supervised contrastive learning [4, 5]. After learning a latent feature space, an inference approach is used to detect OOD samples. This can involve traditional inference approaches such as the Mahalanobis Distance (MD) method [6, 7], or custom approaches that are specifically suited to the latent features learned from contrastive learning. Examples of such approaches include using the MD on several different augmented versions of an image as well as using the cosine similarity or Euclidean distance to the nearest neighbor [8–10]. However, compared to these works, we aim to detect OOD data points by using typicality-based detection, an approach usually used in group OOD detection and that has not previously been used with contrastive learning.

**Contributions** In this paper, we propose a new typicality-based inference method better suited to the point OOD problem. This will exploit $K$-nearest neighbors as a mechanism to automatically generate a group of data points constituted of those closest to a test point in the latent feature space. Appropriate test statistics can then be evaluated for these neighbors as a measure of typicality for the test point under consideration. The key contributions of the paper are therefore as follows:

- We propose a new typicality-based approach for point OOD detection which is used in conjunction with contrastive learning. This method brings the benefit of typicality-based OOD detection to single OOD data point detection (Section 3).
- We analyze the importance of learning latent

*Corresponding Author: yl18410@bristol.ac.uk

Proceedings of the 5th Northern Lights Deep Learning Conference (NLDL), PMLR 233, 2024.

features using contrastive learning for the overall performance of the approach (Section 5.2).

- Empirical evidence is presented to show that the new typicality-based inference approach improves point OOD detection compared to other benchmark inference methods which are also commonly used together with contrastive learning (Section 5.3).

# 2 Background

Throughout this paper, scalar mathematical quantities will be represented in lowercase, and vectors and matrices will be in bold. The term 'class labels' will refer to the labels for the different classes of the ID dataset. The term 'OOD labels' will denote whether the data point is ID or OOD.

## 2.1 Contrastive Learning

Contrastive learning involves assigning an anchor denoted $\boldsymbol{x^{anc}}$, as well as $\boldsymbol{x^{pos}}$ and $\boldsymbol{x^{neg}}$ which denotes an anchor's positive and negative samples respectively. The goal of contrastive loss is to learn a feature vector of $\boldsymbol{x^{anc}}$, denoted $\boldsymbol{z^{anc}} = f_{\boldsymbol{\theta}}(\boldsymbol{x^{anc}})$, similar to the feature vectors of $\boldsymbol{z^{pos}}$ denoted $\boldsymbol{z^{pos}} = f_{\boldsymbol{\theta}}(\boldsymbol{x^{pos}})$, whilst also being dissimilar to $\boldsymbol{z^{neg}}$ denoted $\boldsymbol{z^{neg}} = f_{\boldsymbol{\theta}}(\boldsymbol{x^{neg}})$. The most common form of similarity measure between features $\boldsymbol{z_i}$ and $\boldsymbol{z_j}$ is given by the cosine similarity as given in Eqn. 1:

$$sim(\boldsymbol{z_i}, \boldsymbol{z_j}) = \frac{\boldsymbol{z_i} \cdot \boldsymbol{z_j}}{\|\boldsymbol{z_i}\|\|\boldsymbol{z_j}\|} \tag{1}$$

The type of contrastive loss used in this work is the Supervised Contrastive (SupCLR) loss [5]. In this case, all the data points in the same class as the anchor $\boldsymbol{x^{anc}}$ are treated as positive samples $\boldsymbol{x^{pos}}$ whilst data points in a different class from the anchor are treated as negative samples. The SupCLR contrastive loss is then given by Eqn. 2:

$$\mathrm{L}_{SupCLR} = \frac{-1}{|P(\boldsymbol{z^{anc}})|} \times \sum_{pos \in P(\boldsymbol{z^{anc}})} \log \frac{exp(\boldsymbol{z^{anc}} \cdot \boldsymbol{z^{pos}}/\tau)}{\sum_{i \in A(\boldsymbol{z^{anc}})} exp(\boldsymbol{z^{anc}} \cdot \boldsymbol{z_i}/\tau)} \tag{2}$$

Here, $P(\boldsymbol{z^{anc}})$ is the set of indices of all positives samples of $\boldsymbol{z^{anc}}$ which are present in the batch, and $|P(\boldsymbol{z^{anc}})|$ is its cardinality. $A(\boldsymbol{z^{anc}})$ is the set of all indices in the batch excluding $\boldsymbol{z^{anc}}$ itself, this includes both $\boldsymbol{z^{pos}}$ and $\boldsymbol{z_i}$. Intuitively, the SupCLR loss learns the features in common between different instances of the same class whilst also being invariant to data augmentation.

## 2.2 Typicality

For a group of test inputs of size $K$ denoted $\boldsymbol{X}$, typicality-based methods generally involve examining whether $\boldsymbol{X}$ was sampled from the training distribution $p$ or a different distribution $q$. This is done by using a statistic $\gamma$ and examining its $K$-sample average deviation $\hat{\epsilon}$ from a typical value of the statistic $\hat{\gamma}$, as shown in Eqn. 3:

$$\frac{1}{K}\sum_{l=1}^{K}|(\gamma_l - \hat{\gamma})| = \hat{\epsilon} \tag{3}$$

where $\gamma_l$ denotes the statistic for the $l^{th}$ data point in the group [11]. OOD detection using a typicality-based approach is performed with the assumption that $\hat{\epsilon}$ will be lower for ID data compared to OOD data. This can be problematic in cases where the variance of $\gamma_l$ is high for individual ID data points, making it difficult to determine whether a large $\hat{\epsilon}$ is due to the test data points being OOD or due to the variance in $\gamma_l$ [12]. This is one of the reasons why typicality-based strategies are generally most effective in distinguishing between sets of OOD and ID data points, since the variances of $\gamma_l$ and $\hat{\epsilon}$ tend to be lower when evaluated over larger sets of data points.

# 3 Method

In this section, we introduce a typicality-based OOD detection method that is effective for point OOD detection. The method utilizes latent features learned from contrastive learning in conjunction with the nearest neighbors of a test data point to generate a group of data points. We then calculate how typical an individual data point is based on the calculation of a statistic of interest using the individual data point and its nearest neighbors. We argue that this can be considered as a form of a point OOD detection approach where the nearest neighbors of the test data point provide additional context for deciding whether a single data point is ID or OOD. This differs from a traditional group OOD detection setting in two main ways. First, in group OOD detection the points in a group are usually given, whilst in our case the group is determined from a single data point by using $K$-nearest neighbors. Second, a data point $\boldsymbol{x_1}$ can have another data point $\boldsymbol{x_2}$ as one of its neighbors (i.e. as part of its group) but there is no requirement for $\boldsymbol{x_2}$ to have $\boldsymbol{x_1}$ as one of its nearest neighbors. Our proposed approach consists of the following steps:

1. Pass the given test data point $\boldsymbol{x}$ through a neural network encoder $f$ to obtain its latent features, $\boldsymbol{z} = f(\boldsymbol{x})$.

2. Generate a group of data points for $\boldsymbol{z}$ by determining its $K$-nearest neighbors in the latent space, $\boldsymbol{z_{neighbors}}$.

3. Apply one-dimensional (1D) Typicality inference, which involves calculating a statistic for each of the $K$ data points and quantifying the extent to which they deviate in aggregate from a typical value $\hat{\gamma}$, to obtain an OOD score, $D_{Typical}$.

A schematic for the OOD inference method referred to as '1D Typicality' is shown in Fig. 1.

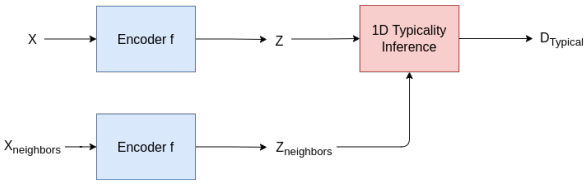



**Figure 1.** Schematic of the 1D Typicality approach. $\boldsymbol{x}$: test image. $\boldsymbol{x_{neighbors}}$: $K$-nearest neighbors of the test input (where distance is measured in latent space). $f$: neural network encoder. $\boldsymbol{z}$: latent features of the test input. $\boldsymbol{z_{neighbors}}$: latent feature of the $K$-nearest neighbors of the test input. $D_{Typical}$: OOD score which uses $\boldsymbol{z}$ and $\boldsymbol{z_{neighbours}}$.

### 3.1 Statistic Selection - Average 1D Mahalanobis Distance

Traditional typicality approaches look at how a single aggregate statistic deviates from the typical value of the statistic. However, a single statistic may be insufficient to separate ID and OOD data. Here we consider using several different statistics to examine the extent to which each deviates from its typical value. The typicality scores related to each statistic are then aggregated to provide a single typicality value for OOD detection. We chose to use a variant of the MD as our statistic as it is a simple and established post-processing method for detecting OOD inputs in neural networks [13]. More specifically, we calculate the one-dimensional Mahalanobis Distance (1D MD) for each different dimension of a test data point and compare it to a typical 1D MD value for each dimension [14]. We calculate the typical values from the training data, where we use the mean value of the 1D MD for each different dimension as the typical values. As is generally the case when evaluating MD, this involves modeling the latent features of the training data using a multivariate Gaussian distribution [13, 15]. After fitting a Gaussian distribution to the data, the linearly independent eigenvectors $\boldsymbol{\nu}$ of the covariance matrix $\boldsymbol{\Sigma}$ can be found. We then calculate the 1D MD for the $d^{th}$ dimension, $MD_{(d)}$, as shown in Eqn. 4:

$$MD_{(d)}(\boldsymbol{z}) = \frac{l_d^2}{\lambda_d} \tag{4}$$

where $\boldsymbol{z}$ is the latent representation of a data point, $\boldsymbol{\mu}$ is the mean of the data in the latent space, $\lambda_d$ is the $d^{th}$ eigenvalue, $l_d = \boldsymbol{\nu_d}^T(\boldsymbol{z} - \boldsymbol{\mu})$ is the projection of $(\boldsymbol{z} - \boldsymbol{\mu})$ to the $d^{th}$ eigenvector $\boldsymbol{v_d}$. $\frac{\boldsymbol{l_d}^2}{\lambda_d}$ is equivalent to the 1D MD from the projected coordinate to the 1D Gaussian distribution $\mathcal{N}(0, \lambda_d)$ [14]. This formulation can be generalized to the case where the data is also classified into different classes, and where the class-specific values for the $i^{th}$ class are given by $MD_{(d,i)}$, $\boldsymbol{\mu_{(i)}}$, $\boldsymbol{\nu_{(d,i)}}$, $\lambda_{(d,i)}$ and $\boldsymbol{l_{(d,i)}}$ for the 1D MD, mean, eigenvectors, eigenvalues and projection vector respectively.

For our work, we use the multi-class formulation where we have class-specific typical values. We use the training data to calculate $\mu_{MD_{(d,i)}}$ and $\sigma_{MD_{(d,i)}}$, the mean and standard deviation, respectively, of the 1D MD values for the $d^{th}$ dimension and $i^{th}$ class. The steps used to compute $\mu_{MD_{(d,i)}}$ and $\sigma_{MD_{(d,i)}}$ are given in Algorithm. 1.

**Algorithm 1** Calculating Typical values - Computing the class-specific mean and standard deviation of the 1D MD value for each dimension

**for** each class $i \in \{1...I\}$ **do**

**Input:** Training samples belonging to class $i$, $\boldsymbol{x_{train(i)}}$

Calculate latent vectors $\boldsymbol{z_{train(i)}}, \boldsymbol{z_{train(i)}} = f(\boldsymbol{x_{train(i)}})$

Calculate Mean $\boldsymbol{\mu_{(i)}}$ and Covariance matrix $\boldsymbol{\Sigma_{(i)}}$ from $\boldsymbol{z_{train(i)}}$.

Calculate $D$ Eigenvectors $\boldsymbol{\nu_{(d,i)}}$ and $D$ Eigenvalues $\lambda_{(d,i)}$ from the Covariance matrix $\boldsymbol{\Sigma_{(i)}}$

**for** each dimension d $\in \{1...D\}$ **do**

Calculate 1D MD, $MD_{(d,i)}$, $MD_{(d,i)}(\boldsymbol{z_{train(i)}}) = \frac{\boldsymbol{\nu_{(d)}}^T(\boldsymbol{z_{train(i)}} - \boldsymbol{\mu_{(i)}})}{\lambda_{(d)}}$

Calculate 1D MD mean $\mu_{MD_{(d,i)}}$ and the 1D MD standard deviation $\sigma_{MD_{(d,i)}}$

**end for**

**end for**

### 3.2 Inference

To perform inference on the latent vector of a test data point $\boldsymbol{z}$, the $K$-nearest neighbors $\boldsymbol{z_{neighbors}}$ of $\boldsymbol{z}$ are found (where when $K = 1$ $\boldsymbol{z_{neighbors}} = \{\boldsymbol{z}\}$). Here we calculate the neighbors based on the other test data points as it removes the requirement of retaining the training data after training the model and calculating the typical values. Additionally, in

a real-world setting, there may not be access to the training data. This is especially important in domains where data privacy is important such as in healthcare [16]. Having found the nearest neighbors, the 1D MD for dimension $d$, class $i$, and member $l$ in the group is calculated as denoted by $MD_{(d,i)}(\boldsymbol{z_l})$, where $\boldsymbol{z_l}$ corresponds to the latent features of the $l^{th}$ member in the group. Finally, an aggregate typicality score $D_{Typical}$ is determined by summing the squares of the deviation of the individual 1D scores from the typical value, $\mu_{MD_{(d,i)}}$ as shown in Eqn. 5. The steps of the inference process are given in Algorithm 2:

$$D_{Typical} = \min_{\mathrm{i}} \sum_{d=1}^{D} \sum_{l=1}^{K} \left[ \frac{MD_{(d,i)}(\boldsymbol{z_l}) - \mu_{MD_{(d,i)}}}{\sigma_{MD_{(d,i)}}} \right]^2 \tag{5}$$

**Algorithm 2** 1D Typicality Inference

**Input:** Test sample $\boldsymbol{x}$

Calculate latent features $\boldsymbol{z}, \boldsymbol{z} = f(\boldsymbol{x})$

Find nearest neighbours of $\boldsymbol{z}$, $\boldsymbol{z_{neighbors}}$, to make a group with $K$ members

**for** each class $i \in \{1 \ldots I\}$ **do**

  **for** each member $l$ in the group **do**

    Calculate 1D Mahalanobis Distance, $MD_{(d,i)}$, $MD_{(d,i)}(\boldsymbol{z_l}) = \frac{\boldsymbol{\nu_{(d)}}^T(\boldsymbol{z_l} - \boldsymbol{\mu_{(i)}})}{\lambda_{(d)}}$

  **end for**

  Calculate typicality score for each class $D_{Typical,i} = \sum_{d=1}^{D} \sum_{l=1}^{K} \left[ \frac{MD_{(d,i)}(\boldsymbol{z_l}) - \mu_{MD_{(d,i)}}}{\sigma_{MD_{(d,i)}}} \right]^2$

**end for**

Calculate lowest class Typicality score, $D_{Typical} = \min_{\mathrm{i}} D_{Typical,i}$

To define a data point as ID and OOD based on the typicality score, we can assume that there is a validation set that was held out from training. A threshold $\alpha$ can be set as a value in which 95% of samples in the validation set satisfy $D_{Typical} < \alpha$, where data points with a $D_{Typical}$ below $\alpha$ are considered ID.

# 4 Experiments

**Datasets** We study OOD detection on the following ID ($D_{in}$) and OOD dataset ($D_{out}$) pairs. We use CIFAR100 [17], Caltech256 [18], and TinyImageNet [19] as our ID datasets while our OOD datasets include the ID datasets as well as CIFAR10 [17], SVHN [20] and MNIST [21]. The procedure involves defining one of the ID datasets as $D_{in}$ and defining one of the OOD datasets as $D_{out}$.

**Metric** We measured the quality of OOD detection using an established metric for this task which is the AUROC. The AUROC measures the Area Under the Receiver Operating Characteristic curve. The Receiver Operating Characteristic (ROC) curve plots the relationship between true positive rate (TPR) and false positive rate (FPR). The area under the ROC curve can be interpreted as the probability that a positive example will have a higher detection score than a negative example. In this case, we treat OOD examples as the positive class. Unless otherwise stated, AUROC values reported in this work are obtained over an average of 8 trials with unique seeds. Additional metrics are also described in Appendix A.1.

**Implementation Details** Experiments were conducted using a Resnet-50 as the encoder $f$ for all the different models and use an output dimensionality of 128 with the outputs being $l_2$ normalized [22]. A Resnet-50 was chosen as the encoder as this is a common architecture used in computer vision, particularly for self-supervised learning. When comparing with other methods that also use contrastive learning-based features, we used the same models trained for our approach and only changed the method of inference. Therefore all the contrastive learning-based feature approaches were trained in the same way, only the inference approach differed. This was done to enable a comparison of the effectiveness of the inference approach without having the model training as a confounding factor. For the models trained with contrastive learning, we used the supervised contrastive learning framework [5]. All models are trained for 300 epochs with a batch size of 256, using the SGD optimizer with a learning rate of $3e^{-2}$, optimizer momentum of 0.9, and weight decay of $1e^{-4}$. The number of neighbors $K$ was chosen to be 10 as explained in Section 5.1. The data augmentations used are similar to those used in [4] but with slightly different parameters due to the different datasets used. See Appendix A.2 for further details.

# 5 Results and Discussion

For the results shown in this section, the performance of OOD detectors varies significantly on different ID-OOD dataset pairs with some AUROC values being below 0.5 whilst others being above 0.9, depending on the difficulty of the task. Although when the AUROC value is close 0.5, the performance of the OOD detector is similar to that of a random detector, comparisons remain valid if they are statistically significant.

## 5.1 Number of Neighbors

To determine the number of neighbors to be used in the algorithm, we compared the AUROC of 1D Typicality for different values of $K$ on the CIFAR100-SVHN ID-OOD dataset pair. Fig 2 shows that performance improves as the number of neighbors $K$ increases, but that the rate of improvement in the AUROC tends to decrease with $K$. This suggests that relatively low $K$ values can be used without incurring a significant loss of performance and that the 1D Typicality approach is not particularly sensitive to the choice of $K$ providing that it is not too small. In the following, we will use a $K$ value of 10 since this tends to achieve good performance whilst also being more computationally efficient compared to larger $K$ values. Results for additional ID-OOD dataset pairs can be seen in Appendix A.3.

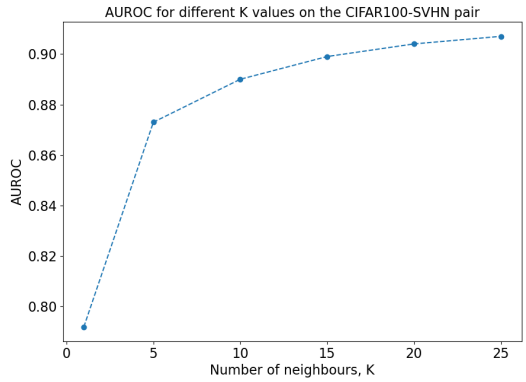



**Figure 2.** AUROC values using the 1D Typicality approach for different numbers of neighbors for the CIFAR100-SVHN ID-OOD dataset pair.

## 5.2 Effect of Contrastive Learning

We investigate the effect of using latent features learned from contrastive learning on the performance of 1D Typicality inference. This was done by comparing 1D Typicality (1D T) with a variation of 1D Typicality which does not use contrastive learning and instead uses latent features learned using the Cross-Entropy (CE) loss. We refer to this variation as 1D Typicality CE (1D T CE). Table 1 shows AUROC scores for different combinations of ID and OOD data. The different rows of the table correspond to different OOD datasets and the three different sections are for the three ID datasets used. As shown in Table 1, when CIFAR100 and Caltech256 are used as the ID dataset, 1D Typicality can consistently outperform 1D Typicality CE. In addition, when TinyImageNet is used as the ID dataset, 1D Typicality can achieve better performance on 3 out of 5 of the ID-OOD dataset pairs. Additionally, for most of the cases where the result of 1D Typicality was better than 1D Typicality CE, the results are shown to be statistically significant as determined by the Wilcoxon signed ranked test where the results below a p-value of 0.05 are significant [23]. This is indicated by * in Table 1. However, it could be seen that the when using CIFAR100 as the ID dataset and TinyImageNet as the OOD dataset, the difference between 1D T CE and 1D T is much larger than the opposite case when TinyImageNet is the ID dataset and CIFAR100 is the OOD dataset. A potential reason for this could be due to the number of true classes in the ID dataset. Due to TinyImageNet having more classes in the ID dataset, it may be unlikely that all the neighbors of a particular data point are in the same class (the purity of the neighbors is less) and therefore the statistics of the group may deviate more from the class-specific typical values of a particular class, which lowers the performance of the approach.

Results which include additional metrics can be seen in Appendix A.5. Overall, these results suggest that using contrastive learning to learn features improves performance compared to CE learned latent features when applying 1D Typicality as the inference method. This is consistent with the findings of [10] which suggest that contrastive learning is beneficial for learning a compact latent space, which aids in using nearest neighbors for OOD detection.

| **Dataset** | **AUROC** | |
|---|---|---|
| | **1D T CE** | **1D T** |
| **ID: CIFAR100** | | |
| OOD: SVHN | 0.473 | **0.898*** |
| OOD: CIFAR10 | 0.460 | **0.755*** |
| OOD: TinyImageNet | 0.512 | **0.779*** |
| OOD: Caltech256 | 0.482 | **0.765*** |
| OOD: MNIST | 0.422 | **0.770*** |
| **ID: Caltech256** | | |
| OOD: SVHN | 0.414 | **0.518*** |
| OOD: CIFAR10 | 0.462 | **0.487*** |
| OOD: CIFAR100 | 0.489 | **0.496*** |
| OOD: TinyImageNet | 0.484 | **0.496*** |
| OOD: MNIST | **0.866** | **0.866** |
| **ID: TinyImageNet** | | |
| OOD: SVHN | **0.587*** | 0.451 |
| OOD: CIFAR10 | 0.488 | **0.526*** |
| OOD: CIFAR100 | 0.506 | **0.513*** |
| OOD: Caltech256 | **0.576*** | 0.548 |
| OOD: MNIST | 0.964 | **0.989*** |

**Table 1.** AUROC for different ID-OOD dataset pairs using 1D Typicality CE ablation as well as 1D Typicality. Bold highlights the best-performing method and * indicates that the best-performing method is statistically significantly better than the other approaches.

## 5.3 Comparing 1D Typicality with other Benchmarks using Contrastive Learning

In this section, we compare 1D Typicality inference with other inference methods which all used latent features trained via supervised contrastive loss [5]. All inference methods used the same network back-

bone architecture and latent dimensionality. More specifically, the inference methods compared include Self-Supervised Outlier Detection (SSD+) [7] and Deep K-Nearest Neighbours (KNN+) [10], both of which have achieved high OOD detection performance compared to other contrastive and non-contrastive baselines. SSD+ is an approach that models the latent space as a multivariate Gaussian distribution for each class, and uses the MD [13] for OOD detection. KNN+ is a non-parametric approach that uses the distance in latent space between a test datapoint and its $K^{th}$ nearest neighbor in the training set to determine if an input is OOD. To enable a direct comparison with our approach, we used the same $K$ value as 1D Typicality ($K = 10$) for the number of neighbors used in KNN+. Results which include additional metrics can be seen in Appendix A.5. The results show that 1D Typicality outperforms SSD+ and KNN+ baselines in the majority of the cases, where the results are also shown to be statistically significant. These results suggest that 1D Typicality inference can outperform other inference methods which also use latent features trained using the same contrastive learning framework.

| Dataset | AUROC | | |
|---|---|---|---|
| | KNN+ | SSD+ | 1D T |
| **ID: CIFAR100** | | | |
| OOD: SVHN | 0.828 | 0.818 | **0.898*** |
| OOD: CIFAR10 | 0.739 | 0.747 | **0.755*** |
| OOD: TinyImageNet | 0.774 | **0.790*** | 0.779 |
| OOD: Caltech256 | 0.733 | **0.768** | 0.765 |
| OOD: MNIST | 0.583 | 0.625 | **0.770*** |
| **ID: Caltech256** | | | |
| OOD: SVHN | **0.535*** | 0.530 | 0.518 |
| OOD: CIFAR10 | 0.479 | 0.483 | **0.487** |
| OOD: CIFAR100 | 0.493 | **0.496** | **0.496** |
| OOD: TinyImageNet | 0.492 | 0.495 | **0.496** |
| OOD: MNIST | 0.554 | 0.500 | **0.866*** |
| **ID: TinyImageNet** | | | |
| OOD: SVHN | 0.360 | 0.288 | **0.451*** |
| OOD: CIFAR10 | 0.521 | 0.518 | **0.526** |
| OOD: CIFAR100 | **0.515** | 0.503 | 0.513 |
| OOD: Caltech256 | 0.512 | 0.538 | **0.548*** |
| OOD: MNIST | 0.921 | 0.963 | **0.989*** |

**Table 2.** AUROC for different ID-OOD dataset pairs using KNN+, SSD+ and 1D Typicality. Bold highlights the best-performing method and * indicates that the best-performing method is statistically significantly better than the other approaches.

### 5.4 Comparing 1D Typicality with Cross-Entropy Baselines

In this section, we compare 1D Typicality with other methods which derive OOD scores from a model trained with the CE loss, and which are commonly used in the literature. This includes Maximum Softmax Probability (MSP) [24], ODIN [25] and the MD method used by [13]. Table 3 shows that 1D Typicality outperforms MSP [24], ODIN [25] and the MD [13] baselines in the majority of the cases. Furthermore, for the case where 1D Typicality does not outperform all the baselines, the performance of 1D Typicality consistently achieves the second-best results. Results including additional metrics can be seen in Appendix A.5.

| Dataset | AUROC | | | |
|---|---|---|---|---|
| | MSP | ODIN | MD | 1D T |
| **ID: CIFAR100** | | | | |
| OOD: SVHN | 0.846 | 0.874 | 0.859 | **0.898*** |
| OOD: CIFAR10 | 0.720 | 0.697 | 0.731 | **0.755*** |
| OOD: TinyImageNet | 0.752 | 0.774 | **0.780** | 0.779 |
| OOD: Caltech256 | 0.707 | 0.716 | 0.756 | **0.765*** |
| OOD: MNIST | 0.533 | 0.567 | 0.651 | **0.770*** |
| **ID: Caltech256** | | | | |
| OOD: SVHN | 0.408 | 0.387 | 0.388 | **0.518*** |
| OOD: CIFAR10 | 0.462 | 0.473 | 0.455 | **0.487*** |
| OOD: CIFAR100 | 0.487 | 0.480 | 0.486 | **0.496*** |
| OOD: TinyImageNet | 0.482 | 0.475 | 0.478 | **0.496*** |
| OOD: MNIST | 0.594 | **0.912*** | 0.711 | 0.866 |
| **ID: TinyImageNet** | | | | |
| OOD: SVHN | 0.310 | 0.450 | **0.566*** | 0.451 |
| OOD: CIFAR10 | 0.499 | 0.485 | 0.487 | **0.526*** |
| OOD: CIFAR100 | 0.476 | 0.493 | 0.502 | **0.513*** |
| OOD: Caltech256 | 0.410 | 0.491 | **0.577*** | 0.548 |
| OOD: MNIST | 0.289 | 0.554 | 0.968 | **0.989*** |

**Table 3.** AUROC for different ID-OOD dataset pairs using baselines trained with the cross-entropy loss and the 1D Typicality approach trained with contrastive learning. Bold highlights the best-performing method and * indicates that the best-performing method is statistically significantly better than the other approaches.

## 6 Conclusion

In this paper, we have proposed a new inference approach that extends typicality-based OOD detection to point OOD detection. We compared our approach to several baselines, and our results showed the following:

- The performance of 1D Typicality is robust to different values of K in K-nearest neighbors providing K is not too low.
- Contrastive learning is crucial for achieving high performance with 1D Typicality.
- By testing on various ID and OOD dataset pairs, we have shown that the 1D Typicality approach achieves better point OOD detection performance than the baselines.

## Acknowledgments

We thank Bahram Keshtmand for useful comments and suggestions. NK is funded by an EPRSC PhD studentship as part of the Centre for Doctoral Training in Future Autonomous and Robotic Systems [grant number EP/L015293/1]. RSR is funded by the UKRI Turing AI Fellowship [grant number EP/V024817/1].

## References


[1] A. Nguyen, J. Yosinski, and J. Clune. "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 427–436. DOI: 10.48550/arXiv.1412.1897.

[2] J. Yang, K. Zhou, Y. Li, and Z. Liu. "Generalized out-of-distribution detection: A survey". In: *arXiv preprint arXiv:2110.11334* (2021). DOI: 10.48550/arXiv.2110.11334.

[3] A. v. d. Oord, Y. Li, and O. Vinyals. "Representation learning with contrastive predictive coding". In: *arXiv preprint arXiv:1807.03748* (2018). DOI: 10.48550/arXiv.1807.03748.

[4] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. "A simple framework for contrastive learning of visual representations". In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607. DOI: 10.48550/arXiv.2002.05709.

[5] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. "Supervised contrastive learning". In: *arXiv preprint arXiv:2004.11362* (2020). DOI: 10.48550/arXiv.2004.11362.

[6] J. Winkens, R. Bunel, A. G. Roy, R. Stanforth, V. Natarajan, J. R. Ledsam, P. MacWilliams, P. Kohli, A. Karthikesalingam, S. Kohl, et al. "Contrastive training for improved out-of-distribution detection". In: *arXiv preprint arXiv:2007.05566* (2020). DOI: 10.48550/arXiv.2007.05566.

[7] V. Sehwag, M. Chiang, and P. Mittal. "Ssd: A unified framework for self-supervised outlier detection". In: *arXiv preprint arXiv:2103.12051* (2021). DOI: 10.48550/arXiv.2103.12051.

[8] J. Tack, S. Mo, J. Jeong, and J. Shin. "Csi: Novelty detection via contrastive learning on distributionally shifted instances". In: *arXiv preprint arXiv:2007.08176* (2020). URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/8965f76632d7672e7d3cf29c87ecaa0c-Paper.pdf.

[9] H. Cho, J. Seol, and S.-g. Lee. "Masked Contrastive Learning for Anomaly Detection". In: *arXiv preprint arXiv:2105.08793* (2021). DOI: 10.48550/arXiv.2105.08793.

[10] Y. Sun, Y. Ming, X. Zhu, and Y. Li. "Out-of-distribution detection with deep nearest neighbors". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 20827–20840. DOI: 10.48550/arXiv.2204.06507.

[11] E. Nalisnick, A. Matsukawa, Y. W. Teh, and B. Lakshminarayanan. "Detecting out-of-distribution inputs to deep generative models using typicality". In: *arXiv preprint arXiv:1906.02994* 5 (2019), p. 5. DOI: 10.48550/arXiv.1706.02690.

[12] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[13] K. Lee, K. Lee, H. Lee, and J. Shin. "A simple unified framework for detecting out-of-distribution samples and adversarial attacks". In: *Advances in neural information processing systems* 31 (2018). URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/abdeb6f575ac5c6676b747bca8d09cc2-Paper.pdf.

[14] J. Ren, S. Fort, J. Liu, A. G. Roy, S. Padhy, and B. Lakshminarayanan. "A Simple Fix to Mahalanobis Distance for Improving Near-OOD Detection". In: *arXiv preprint arXiv:2106.09022* (2021). DOI: 10.48550/arXiv.2106.0902.

[15] X. Du, Z. Wang, M. Cai, and Y. Li. "VOS: Learning What You Don't Know by Virtual Outlier Synthesis". In: *arXiv preprint arXiv:2202.01197* (2022). DOI: 10.48550/arXiv.2202.01197.

[16] B. Murdoch. "Privacy and artificial intelligence: challenges for protecting health information in a new era". In: *BMC Medical Ethics* 22.1 (2021), pp. 1–5. DOI: 10.1186/s12910-021-00687-3.

[17] A. Krizhevsky, G. Hinton, et al. "Learning multiple layers of features from tiny images". In: (2009). URL: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[18] G. Griffin, A. Holub, and P. Perona. "Caltech-256 object category dataset". In: (2007).

[19] Y. Le and X. S. Yang. "Tiny ImageNet Visual Recognition Challenge". In: 2015. URL: http://vision.stanford.edu/teaching/cs231n/reports/2015/pdfs/yle_project.pdf.

[20] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. "Reading digits in natural images with unsupervised feature learning". In: (2011). URL: http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

[21] Y. LeCun, C. Cortes, and C. Burges. "MNIST handwritten digit database". In: *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist* 2 (2010).

[22] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778. DOI: 10.48550/arXiv.1512.03385.

[23] A. Durango and C. Refugio. "An Empirical Study on Wilcoxon Signed Rank Test". In: *J. Negros Orient. State Univ.,(December)* (2018).

[24] D. Hendrycks and K. Gimpel. "A baseline for detecting misclassified and out-of-distribution examples in neural networks". In: *arXiv preprint arXiv:1610.02136* (2016). DOI: 10.48550/arXiv.1610.02136.

[25] S. Liang, Y. Li, and R. Srikant. "Enhancing the reliability of out-of-distribution image detection in neural networks". In: *arXiv preprint arXiv:1706.02690* (2017). DOI: 10.48550/arXiv.1706.02690.

# A Appendix

## A.1 Additional Metrics

In addition to the AUROC, we measured the quality of OOD detection using other established metrics for this task, which are the FPR at 95% TPR, and the AUPR [24].

- **FPR at 95% TPR:** Measures the false positive rate (FPR) when the true positive rate (TPR) is equal to 95%. Let TP, FP, TN, and FN represent true positives, false positives, true negatives, and false negatives, respectively. The false positive rate is calculated as FPR=FP/(FP+TN), while the true positive rate is calculated as TPR=TP/(TP+FN).

- **AUPR** Measures the Area Under the Precision - recall (PR) curve. The PR curve is obtained by plotting precision = TP/(TP + FP) versus recall = TP/(TP + FN).

For these metrics, we treat OOD examples as the positive class. These additional metrics are used when showing the full results of the different experiments conducted as part of the main paper.

## A.2 Data augmentation

The data augmentations used are similar to those used in [4] but with slightly different parameters due to the different datasets used. We resize all data points to a height and width of 32 and then utilize random crop (with resize and random flip), random color distortions, and random Gaussian blur as the data augmentations. The details are as follows:

- **Random Crop and Resize:** We use random cropping with a crop of random size (uniform from 0.2 to 1.0 in the area) of the original size followed by rescaling the crop to the original size of the image. This has been implemented in Pytorch using 'torchvision.transforms.RandomResizedCrop'. Additionally, the random crop (with resize) is always followed by a random horizontal flip with a 50 % probability.

- **Color Distortion:** Color distortion is composed of color jittering and color dropping. Color jittering was implemented in Pytorch using 'torchvision. transforms. ColorJitter' with brightness, contrast, saturation, and hue values of 0.4, 0.4, 0.4, and 0.1 respectively. Color jittering was followed by a color dropping by converting the image to grayscale with a 20 % probability. Color dropping was implemented in Pytorch using 'torchvision.transforms.RandomGrayscale'.

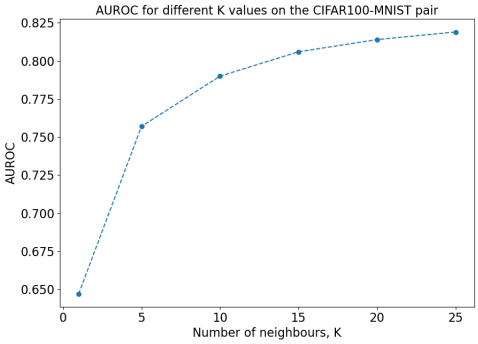
- **Gaussian Blur:** We blur the image 50 % of the time using a Gaussian kernel which was implemented in Pytorch using 'torchvision.transforms.GaussianBlur' with a $\sigma \in [0.1, 2.0]$.

## A.3 Choosing the Number of Neighbors

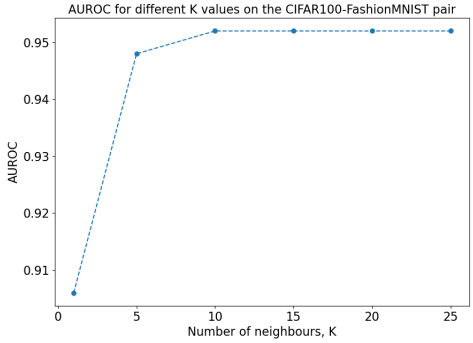
In addition to Fig 2, we also compared the AUROC of 1D Typicality for different values of $K$ on other ID-OOD dataset pairs where CIFAR100 is the ID dataset and MNIST, FashionMNIST, KMNIST OOD datasets. Figs A.1(a) - A.1(c) agreed with results shown in 2 as we can see that the performance improved as the number of neighbors $K$ increases, but that the rate of improvement in the AUROC tends to decrease with $K$, therefore showing that 1D Typicality is not particularly sensitive to the choice of $K$ providing that it is not too small.

## A.4 The OOD Label of Neighbors

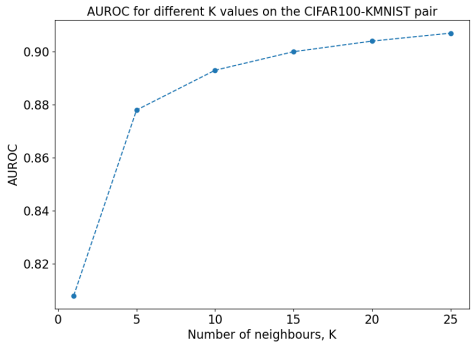
A key assumption underpinning the 1D Typicality method is that an ID data point will tend to have ID neighbors whilst conversely an OOD data point will tend to have OOD neighbors. To test




**(a)** MNIST




**(b)** FashionMNIST




**(c)** KMNIST

**Figure A.1.** AUROC values using the 1D Typicality approach for different numbers of neighbors on a variety of OOD datasets when CIFAR100 is used as the ID dataset.

this assumption we calculated the distance between the latent vectors of all the test data points (ID and OOD) and then evaluated the percentage of the nearest 10 neighbors which were OOD for the ID and OOD data points. For an ID data point, we refer to the percentage of neighbors which were OOD as ID-OOD. Similarly, for an OOD data point, we refer to the percentage of neighbors which were OOD as OOD-OOD. The ID-OOD and OOD-OOD for all the data points in the ID and OOD datasets were then averaged resulting in average ID-OOD and OOD-OOD as shown in Table A.1. Here we use CIFAR100 as the ID dataset and MNIST, FashionMNIST, KMNIST and SVHN as OOD datasets. For each ID-OOD dataset pair we used 10,000 ID and OOD data points. Table A.1 shows that ID-OOD was below 10% when MNIST, FashionMNIST and KMNIST were used as OOD datasets. This suggests that for the majority of the CIFAR100 ID data points, 9 or 10 of their nearest neighbors were also ID data. In the case where the OOD dataset was SVHN, ID-OOD was 12.32%. This further supports the fact that a majority of the CIFAR100 data points were also ID. Additionally, it can be seen that for all 4 OOD datasets, the OOD-OOD is above 89% indicating that most of the OOD data points have mostly OOD neighbors. Overall, these results support the assumption that ID (OOD) data points will tend to have mostly ID (OOD) neighbors at least for the datasets considered.

| OOD Dataset | ID-OOD (%) | OOD-OOD (%) |
|---|---|---|
| MNIST | 2.13 | 99.24 |
| FashionMNIST | 6.10 | 89.91 |
| KMNIST | 2.82 | 98.33 |
| SVHN | 12.32 | 94.43 |

**Table A.1.** Average ID-OOD and OOD-OOD percentage values for different OOD datasets when CIFAR100 is used as the ID dataset.

## A.5 Full Results

| Dataset | AUROC | AUPR | FPR |
|---|---|---|---|
| | 1D T CE/1D T | | |
| | **ID: CIFAR100** | | |
| OOD: SVHN | 0.473 / **0.898*** | 0.698 / **0.941*** | 0.874 / **0.311*** |
| OOD: CIFAR10 | 0.460 / **0.755*** | 0.458 / **0.711*** | 0.918 / **0.710*** |
| OOD: TinyImageNet | 0.512 / **0.779*** | 0.504 / **0.748*** | 0.925 / **0.682*** |
| OOD: Caltech256 | 0.482 / **0.765*** | 0.479 / **0.748*** | 0.923 / **0.787*** |
| OOD: MNIST | 0.422 / **0.770*** | 0.442 / **0.753*** | 0.857 / **0.561*** |
| | **ID: Caltech256** | | |
| OOD: SVHN | 0.414 / **0.518*** | 0.673 / **0.731*** | 0.975 / **0.946*** |
| OOD: CIFAR10 | 0.462 / **0.487*** | 0.467 / **0.484*** | 0.956 / **0.953** |
| OOD: CIFAR100 | 0.489 / **0.496*** | **0.499** / 0.498 | 0.956 / **0.951*** |
| OOD: TinyImageNet | 0.484 / **0.496*** | 0.490 / **0.496** | 0.953 / **0.951** |
| OOD: MNIST | **0.866** / **0.866** | 0.830 / **0.852** | 0.486 / **0.450** |
| | **ID: TinyImageNet** | | |
| OOD: SVHN | **0.587** / 0.451 | **0.765** / 0.685 | **0.900** / 0.953 |
| OOD: CIFAR10 | 0.488 / **0.526*** | 0.478 / **0.522*** | **0.949** / 0.951 |
| OOD: CIFAR100 | 0.506 / **0.513*** | 0.504 / **0.516*** | **0.950** / 0.951 |
| OOD: Caltech256 | **0.576** / 0.548 | **0.563** / 0.534 | **0.920** / 0.938 |
| OOD: MNIST | 0.964 / **0.989*** | 0.930 / **0.966*** | 0.086 / **0.022*** |

**Table A.2.** AUROC, AUPR and FPR for different ID-OOD dataset pairs using 1D Typicality CE ablation as well as 1D Typicality. Bold highlights the best-performing method and * indicates that best-performing method is statistically significantly better than the other approaches.

| Dataset | AUROC | AUPR | FPR |
|---|---|---|---|
| | KNN+/SSD+/1D T | | |
| | **ID: CIFAR100** | | |
| OOD: SVHN | 0.828 / 0.818 / **0.898*** | 0.897 / 0.870 / **0.941*** | 0.483 / 0.512 / **0.311*** |
| OOD: CIFAR10 | 0.739 / 0.747 / **0.755*** | 0.696 / 0.698 / **0.711*** | 0.705 / **0.698** / 0.710 |
| OOD: TinyImageNet | 0.774 / **0.790** / 0.779 | 0.734 / **0.749** / 0.748 | 0.641 / **0.611** / 0.682 |
| OOD: Caltech256 | 0.733 / **0.768** / 0.765 | 0.706 / 0.747 / **0.748*** | 0.793 / **0.772** / 0.787 |
| OOD: MNIST | 0.583 / 0.625 / **0.770*** | 0.558 / 0.601 / **0.753*** | 0.803 / 0.810 / **0.561*** |
| | **ID: Caltech256** | | |
| OOD: SVHN | **0.535** / 0.530 / 0.518 | **0.743** / 0.740 / 0.731 | **0.934** / 0.939 / 0.946 |
| OOD: CIFAR10 | 0.479 / 0.483 / **0.487** | 0.480 / **0.484** / **0.484** | **0.952** / 0.954 / 0.953 |
| OOD: CIFAR100 | 0.493 / **0.496** / **0.496** | 0.495 / 0.497 / **0.498** | 0.953 / 0.952 / **0.951** |
| OOD: TinyImageNet | 0.492 / 0.495 / **0.496** | 0.494 / **0.497** / 0.496 | 0.953 / **0.951** / **0.951** |
| OOD: MNIST | 0.554 / 0.500 / **0.866*** | 0.549 / 0.501 / **0.852*** | 0.887 / 0.935 / **0.450*** |
| | **ID: TinyImageNet** | | |
| OOD: SVHN | 0.360 / 0.288 / **0.451*** | 0.658 / 0.605 / **0.685*** | 0.992 / 0.994 / **0.953*** |
| OOD: CIFAR10 | 0.521 / 0.518 / **0.526** | 0.515 / 0.511 / **0.522*** | **0.939** / 0.948 / 0.951 |
| OOD: CIFAR100 | **0.515** / 0.503 / 0.513 | **0.520** / 0.505 / 0.516 | 0.956 / **0.950** / 0.951 |
| OOD: Caltech256 | 0.512 / 0.538 / **0.548*** | 0.527 / 0.530 / **0.534** | 0.983 / 0.948 / **0.938*** |
| OOD: MNIST | 0.921 / 0.963 / **0.989*** | 0.912 / 0.926 / **0.966** | 0.283 / 0.096 / **0.022*** |

**Table A.3.** AUROC, AUPR and FPR for different ID-OOD dataset pairs using KNN+, SSD+ and 1D Typicality. Bold highlights the best-performing method and * indicates that best-performing method is statistically significantly better than the other approaches.

| Dataset | AUROC | AUPR | FPR |
|---|---|---|---|
| | MSP/ODIN/MD/1D T | | |
| | **ID: CIFAR100** | | |
| OOD: SVHN | 0.846 / 0.874 / 0.859 / **0.898*** | 0.926 / 0.940 / 0.901 / **0.941** | 0.507 / 0.447 / 0.466 / **0.311*** |
| OOD: CIFAR10 | 0.720 / 0.697 / 0.731 / **0.755*** | 0.678 / 0.661 / 0.682 / **0.711*** | 0.721 / 0.840 / 0.734 / **0.710** |
| OOD: TinyImageNet | 0.752 / 0.774 / **0.780** / 0.779 | 0.719 / 0.741 / 0.740 / **0.748*** | 0.686 / 0.661 / **0.657** / 0.682 |
| OOD: Caltech256 | 0.707 / 0.716 / 0.756 / **0.765*** | 0.681 / 0.693 / 0.735 / **0.748*** | 0.840 / 0.830 / 0.813 / **0.787*** |
| OOD: MNIST | 0.533 / 0.567 / 0.651 / **0.770*** | 0.525 / 0.562 / 0.621 / **0.753*** | 0.821 / 0.800 / 0.766 / **0.561*** |
| | **ID: Caltech256** | | |
| OOD: SVHN | 0.408 / 0.387 / 0.388 / **0.518*** | 0.675 / 0.656 / 0.656 / **0.731*** | 0.979 / 0.964 / 0.976 / **0.946*** |
| OOD: CIFAR10 | 0.462 / 0.473 / 0.455 / **0.487*** | 0.475 / 0.478 / 0.463 / **0.484** | 0.969 / **0.951** / 0.960 / 0.953 |
| OOD: CIFAR100 | 0.487 / 0.480 / 0.486 / **0.496*** | 0.495 / 0.489 / 0.497 / **0.498** | 0.969 / 0.954 / 0.957 / **0.951** |
| OOD: TinyImageNet | 0.482 / 0.475 / 0.478 / **0.496*** | 0.492 / 0.486 / 0.487 / **0.496** | 0.969 / 0.954 / 0.957 / **0.951** |
| OOD: MNIST | 0.594 / **0.912** / 0.711 / 0.866 | 0.561 / **0.868** / 0.694 / 0.852 | 0.895 / **0.258** / 0.769 / 0.450 |
| | **ID: TinyImageNet** | | |
| OOD: SVHN | 0.310 / 0.450 / **0.566** / 0.451 | 0.611 / 0.679 / **0.750** / 0.685 | 0.978 / 0.937 / **0.911** / 0.953 |
| OOD: CIFAR10 | 0.499 / 0.485 / 0.487 / **0.526*** | 0.491 / 0.483 / 0.478 / **0.522*** | **0.936** / 0.954 / 0.948 / 0.951 |
| OOD: CIFAR100 | 0.476 / 0.493 / 0.502 / **0.513*** | 0.484 / 0.494 / 0.501 / **0.516*** | 0.962 / 0.953 / **0.951** / **0.951** |
| OOD: Caltech256 | 0.410 / 0.491 / **0.577** / 0.548 | 0.452 / 0.502 / **0.563** / 0.534 | 0.983 / 0.957 / **0.920** / 0.938 |
| OOD: MNIST | 0.289 / 0.554 / 0.968 / **0.989*** | 0.424 / 0.579 / 0.941 / **0.966*** | 0.968 / 0.945 / 0.090 / **0.022*** |

**Table A.4.** AUROC, AUPR and FPR for different ID-OOD dataset pairs using baselines trained with the cross-entropy loss and the 1D Typicality approach trained with contrastive learning. Bold highlights the best-performing method and * indicates that 1D Typicality is significantly better than all baselines.