Gaussian Latent Representations for Uncertainty Estimation using Mahalanobis Distance in Deep Classifiers

Aishwarya Venkataramanan\textsuperscript{1,2,3} Assia Benbihi\textsuperscript{4} Martin Laviale\textsuperscript{1,3} Cédric Pradalier\textsuperscript{2,3}

\textsuperscript{1}Université de Lorraine, CNRS, LIEC, Metz, France
\textsuperscript{2}Georgia Tech Europe, GT-CNRS IRL 2958, Metz, France
\textsuperscript{3}LTSER-“Zone Atelier Moselle”, Metz, France
\textsuperscript{4}Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague

Abstract

Recent works show that the data distribution in a network’s latent space is useful for estimating classification uncertainty and detecting Out-Of-Distribution (OOD) samples. To obtain a well-regularized latent space that is conducive for uncertainty estimation, existing methods bring in significant changes to model architectures and training procedures. In this paper, we present a lightweight and high-performance regularization method for Mahalanobis distance (MD)-based uncertainty prediction, and that requires minimal changes to the network’s architecture. To derive Gaussian latent representation favourable for MD calculation, we introduce a self-supervised representation learning method that separates in-class representations into multiple Gaussians. Classes with non-Gaussian representations are automatically identified and dynamically clustered into multiple new classes that are approximately Gaussian. Evaluation on standard OOD benchmarks shows that our method achieves state-of-the-art results on OOD detection and is very competitive on predictive probability calibration. Finally, we show the applicability of our method to a real-life computer vision use case on microorganism classification.

1. Introduction

Current deep learning classification networks achieve superior performance and find widespread applications in various industrial domains such as biology and robotics [16, 30, 41]. While they achieve state-of-the-art accuracy, there remain two main challenges that hinder the deployment of deep classifiers in critical situations: the derivation of calibrated classification and a measure of the classification uncertainty. Without those, a network exposed to Out-of-Distribution (OOD) data makes incorrect predictions with high confidence [17] and no human-in-the-loop can catch such errors. It is thus necessary to obtain calibrated probabilities [17] i.e., predict probabilities that represent true likelihood, and to estimate the uncertainty in the network’s predictions to allow users to make informed decisions.

Among deep uncertainty estimation approaches [1, 12, 14, 21] are Bayesian Neural Networks [2], MC-Dropout [13] and Deep Ensemble [26]. These stochastic methods require multiple forward-passes, so they are not scalable to large systems. Aware of the scalability requirements, current research focuses on estimating uncertainty from deterministic single-forward-pass networks [19, 32, 35, 38, 42, 45]. Distance-based methods belong to this category and are an attractive alternative for their excellent performance in OOD detection [29, 46].

Distance-based methods rely on the distance between the test samples and the In-Distribution (ID) samples in a network’s latent space to determine if the test samples...
We design the method that constrains a classification network’s latent space to fulfill such an assumption [7]. Instead, research on representation learning shows that each class is usually composed of several clusters of visually similar images [3, 9, 50]. This can be due to intra-class variance of images taken from different viewpoints, the presence of additional objects in the image, and variations in object shapes. In the network’s latent space, these variations appear as distinct distributions or deviate from a Gaussian distribution. This breaks the MD assumption, which could lead to incorrect or imprecise uncertainty estimation. In this paper, we introduce MAPLE, a self-supervised representation learning method that regularizes a classification network’s latent space to exhibit multivariate Gaussian distributions. MAPLE generates a latent space where class representations are Gaussian, making it compliant with the MD assumption and allows fast and high-performance MD-based OOD detection, uncertainty estimation, and calibrated classification. The effect of MAPLE is illustrated in Fig. 1 with the 2D projection of the latent space of a Convolutional Neural Network (CNN) trained on CIFAR10.

MAPLE stands for MAhalanobis distance based uncertainty Prediction for reLiablE classification, and is illustrated in Fig. 2. MAPLE relies on two components: i) a self-supervised intra-class label refinement through clustering in the latent space; ii) a deep metric learning loss that improves the class separation. During training, the representations associated to a class that deviate from a Gaussian distribution are divided into several clusters that are approximately Gaussian. The cluster assignments become the new labels of the representations, and the training goes on. Since each cluster gathers samples that exhibit similar intra-class variations, the clustering step is akin to automatic fine-grained annotation. The metric-learning then reinforces the fined-grained class separation by pushing apart the new classes. The combination of in-class clustering and metric learning results in classification representations that are well-clustered and approximately Gaussian, which makes them suitable for MD-based uncertainty estimation.

We evaluate MAPLE against existing uncertainty quantification methods on the following standard benchmarks: CIFAR10 [25] vs. SVHN [36], CIFAR100 [25], CIFAR100 vs. CIFAR10/Tiny ImageNet [28], ImageNet [40] vs. ImageNet-O [20] for OOD detection and predictive probability calibration. Results show that MAPLE achieves the best compromise between performance and run time efficiency while being the most lightweight integration-wise. Also, it introduces minor architectural changes and does not require additional fine-tuning to OOD datasets.

We summarize the paper’s contributions as follows. i) We develop a self-supervised representation learning method that constrains a classification network’s latent space to be approximately Gaussian. ii) We show that such representations allow for reliable OOD detection and probability calibration using MD. iii) We design the method such that it has a minimal impact on the network’s original architecture, and achieves results competitive with the state-of-the-art on OOD detection. The code is available in: https://github.com/vaishwarya96/MAPLE-uncertainty-estimation.git

2. Related Work

Multi-forward-pass Uncertainty Estimation. Traditional uncertainty quantification methods rely on Bayesian Neural Networks [15, 23] to learn a distribution over the network weights. To extract predictive probability variance, sampling [5] or variational methods [2] are used. The application of these methods is limited, as they increase the number of parameters by a factor of two and hinder convergence. As a lighter alternative, MC Dropout [13] enables dropout at test time and averages the network’s output over several forward passes. While MC Dropout paves the way towards faster and lighter uncertainty estimation, it has been shown to produce over-confident predictions [26] and underestimate uncertainty [43]. To improve uncertainty estimation, Deep Ensembles [26] average the predictions from an ensemble of trained models and achieve state-of-the-art performance on several classification tasks. It remains com-
putationally expensive due to the training of multiple models and the several forward passes during inference. By deriving uncertainty from a single forward pass, MAPLE achieves significantly faster inference time without sacrificing performance.

**Single-forward-pass Uncertainty Estimation.** One line of work relies on the distribution of data samples in the network’s latent space. A test sample is considered ID if it lies within the training data manifold, otherwise it is labelled as OOD. Methods differ in the way they regularize the representation space and the way they derive distances. DUQ [39] uses a Radial Basis Function (RBF) kernel in the representation space to measure distances between test samples and the centroids of various classes. Additionally, they use gradient penalty to obtain a regularized space, which improves the prediction’s quality. SNGP [31] uses Spectral Normalization on the network’s weights to satisfy the bi-Lipchitz condition, which is a more gradient-friendly regularization than DUQ. This condition preserves semantically meaningful distance changes in the representation space with respect to input changes. The prediction’s uncertainty is then given by a Gaussian Process layer on the output. To improve the scalability of the Gaussian Process estimation, [45] proposes Deep Kernel Learning to process the input images with a distance-preserving network and fit a Gaussian on inducing points only. Contrary to these methods, MAPLE avoids the Gaussian Process estimation and gradient regularization during training and instead relies on simple metric learning. Similarly, [6] and VMDLS [7] simplifies the Gaussian enforcement by training the network with a Bregman divergence and KL-divergence loss respectively, so that each class representations follow an isotropic Gaussian distribution in the latent space for OOD detection. However, this ignores the possible intra-class variation within each class and requires the Gaussian variance to be tuned manually. Instead, MAPLE uses a simpler self-supervised clustering that automatically fits the data. Also, MAPLE makes the latent space not only suitable for OOD detection but also for calibrated probability prediction.

**Mahalanobis-Distance for OOD detection.** MD is a common distance in the OOD detection literature. Early work by Lee et al. [29] derives confidence values as a function of MD to predict the likelihood of a sample being ID. To obtain competitive performance, the method requires several tweaks such as adding noise to input samples, combining confidence values from multiple feature layers, and fine-tuning on OOD datasets. [24] proposes two light improvements: Partial MD and Marginal MD. In Partial MD, the MD is computed on lower dimensional representations with PCA. Marginal MD uses all training representations to fit a single Gaussian to calculate the MD. While both perform well on Far-OOD datasets i.e., where ID and OOD samples are significantly distinct, their results are limited on Near-OOD [11], where the OOD samples are semantically similar to the ID ones. Relative MD (RMD) [39] improves the MD performance on Near-OOD by computing a global MD between the test sample and the samples of all classes combined, and then subtracting this value from the per-class MDs. All these methods exhibit satisfying performance, but their main limitation is their strong assumption that the image representations follow a Gaussian distribution, even though standard classification training does not enforce such a constraint. MAPLE addresses this limitation with a self-supervised regularization. By doing so, the features better fit the theoretical framework of MD-based OOD detection, thereby improving the performance.

### 3. Method

In this section, we describe MAPLE, a self-supervised regularization method for MD-based OOD detection, uncertainty estimation, and calibrated classification. It augments a standard CNN classifier with a self-supervised regularization to output both class probabilities and MD-based uncertainty. To enable MD for OOD detection, the representations of the training samples are dynamically clustered into multiple Gaussians using X-Means [37] during training. The samples are assigned new pseudo-class labels defined by their cluster assignment. The network is then optimized with the cross-entropy loss and the triplet loss. With periodic validation, the clusters are updated and the total number of classes change with every validation. At inference time, the MD between a test sample and each cluster’s centroid is used to estimate the classification uncertainty and the probability of the point being OOD. Note that the only modification to the original network architecture is in the final layer, where the number of output neurons change according to the number of clusters identified. This makes MAPLE easy to integrate to any classification network. An algorithmic and computational description is provided in Appendix C.

#### 3.1. Representation Regularization

This section describes the self-supervised automatic label refinement through clustering.

**Notations.** Consider a classification problem consisting of $k$ classes with input samples $x$ and labels $y \in \{1, \ldots, k\}$. The training dataset is denoted by $\mathcal{D}_{\text{train}} = \{(x_n, y_n)\}_{n=1}^N$ and the validation dataset by $\mathcal{D}_{\text{val}} = \{(x_m, y_m)\}_{m=N+1}^M$. Let the training input samples be represented as $x_{\text{train}} = \{x_n\}_{n=1}^N$. The training is done with an off-the-shelf classification CNN. The penultimate layer of the CNN is used as a deep feature extractor $f^\theta(\cdot)$, where $f^\theta : \mathbb{R}^D \rightarrow \mathbb{R}^d$ is a mapping from an input of dimension $D$ to a representation (or feature) of dimension $d$, and $\theta$ is the model’s parameters. The final layer consists of $k$ neurons, followed by softmax activation to obtain the
predictive probabilities. In addition to the standard cross-entropy loss used in CNNs, we use the triplet loss on the representations to train the network. See Appendix. B.6 for a recall of these standard losses.

Self-Supervised Dynamic Relabelling. During training, MAPLE updates the training labels to make them representative of the features’ separation in the latent space. Every $p$ epochs, the network is evaluated on $D_{val}$ and the classes with a false negative ratio higher than a threshold $t$ are updated. This is representative of the scenarios where the samples of a given class are misclassified, which is typical of classes with high intra-class variations. For every class to update the training representations belonging to such a class are extracted and clustered using X-Means [37]. The resulting clusters form well-separated groups and we use the cluster assignment as new pseudo-labels for the train samples. If $k'$ additional clusters are introduced by X-Means, each of them are considered as independent classes. Thus, the number of classes becomes $K = k + k'$, and the final layer of the model is updated to have $K$ neurons. Then, the network training continues with the new labels. During inference, the pseudo labels are remapped to the original set of $k$ labels to identify their original class.

Fig. 3 illustrates the benefits of jointly using X-Means and the triplet loss on the representations: X-Means splits classes with high intra-class variations into separated classes that are semantically more representative of the data, and the triplet loss reinforces this separation.

The method introduces three hyperparameters: false negative ratio threshold $t$, frequency of validation epochs $p$ and the maximum number of clusters (max_num_cluster), which is a parameter needed for X-Means. More details on the hyperparameters are provided in Appendix. B.5.

Clustering. The motivation for using X-Means over other commonly used clustering methods such as K-Means [33], DB-SCAN [10] and Gaussian Mixture Models (GMMs) are two-folds: (1) X-Means is scalable and automatically identifies the number of clusters based on the Bayesian Information Criterion (BIC); (2) BIC uses a maximum likelihood estimation of the variance under the spherical Gaussian assumption, which means that the samples are approximately spherical Gaussian in each cluster.

3.2. Representation Distance

This section describes the MD derivation over the latent representations. To avoid matrix singularities, the latent representations are first reduced using PCA.

Dimensionality reduction. Representations extracted from large neural networks usually have a high dimension and redundant dimensions. The MD requires calculating the inverse covariance matrix of these features, but the presence of redundancy causes the covariance matrix to be singular. Furthermore, [39] shows that the presence of non-informative dimensions could be detrimental to MD performance. This motivates the use of dimensionality reduction.

A common dimensionality reduction method is t-SNE [47], widely used for latent space’s visualization. While t-SNE maintains the local distribution of points, it fails to represent global distributions accurately, which is undesirable in distance-based uncertainty predictions. Instead, we use Principal Component Analysis (PCA) for dimensionality reduction. The principal components are constructed from the covariance matrix of the standardized training representations. The eigen vectors of the covariance matrix are the principal components and the eigen values account for the amount of original information (variance) present in these components. We automatically estimate the number of principal components by the number of eigen values in decreasing order, required to explain 95% of the original data variance. This transformation is denoted by $g: \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$, where $d'$ is the dimension of the reduced features. With $x'_{\text{train}} = f^\theta(x_{\text{train}})$ the full dimensions training features, we denote $z'_{\text{train}} = g(x'_{\text{train}})$ the reduced features.

Mahalanobis Distance. The MD is a generalized version of Euclidean distance that takes into account the data correlation to measure the distance. Hence, the MD is more accurate when predicting the distance between a point and a distribution of points. Here, MD is calculated on the PCA-reduced representations as follows. Let $\{z_i\}$ be the set of training representations after dimensionality reduction, $\mu_c$ be the class centroids with $c = 1, 2, ..., K$, and $\Sigma$ be the shared covariance for all training samples, given by

$$
\mu_c = \frac{1}{N_c} \sum_{i: y_i = c} z_i,
$$

$$
\Sigma = \frac{1}{N} \sum_c \sum_{i: y_i = c} (z_i - \mu_c)(z_i - \mu_c)^T.
$$

The following Eq. 2 gives the Mahalanobis distance between the centroid $\mu_c$ of class $c$ and a test sample $\hat{x}$ with
reduced representation $\tilde{z} = g(f^\theta(\tilde{x}))$

$$MD_c(\tilde{x}) = \sqrt{(\tilde{z} - \mu_c)^T \Sigma^{-1}(\tilde{z} - \mu_c)}$$ (2)

3.3. Classification and Uncertainty Estimation

We now show how to use the MD distance calculated in Eq. 2 for three purposes: classification, predictive probability, and uncertainty prediction.

**MD-based Classification.** The predicted class is the one whose centroid $c^*$ is closest to the test sample $\tilde{x}$:

$$c^* = \arg\min_c (MD_c(\tilde{x}))$$ (3)

Note that this classification is inferred in addition to the usual classification done by the network by taking the maximum of the output logits.

**Predictive Probability.** We convert the MD into a calibrated classification probability using the following property: the squared MD on representations with dimension $d^2$ follows a chi-squared distribution $\chi^2_{d^2}$ with $d^2$ degrees of freedom. The proof of this is provided in Appendix F. The MD is converted as follows:

$$P^c_{MD} = 1 - \text{cdf}(\chi^2_{d^2}(MD_c(\tilde{x})^2))$$ (4)

where $\text{cdf}(.)$ is the cumulative distribution function. $P^c_{MD}$ represents the probability that a test sample belongs to class $c$. When the test point belongs to a particular class, the MD to that class is low and the corresponding $P^c_{MD}$ is high. The predictive probability is the one associated with the class $c^*$ obtained in Eq. 3:

$$P^{c^*}_{MD} = \max_c (P^c_{MD})$$ (5)

Note that contrary to a CNN softmax ‘probabilities’, this classification probability is calibrated and can be interpreted as a confidence in the classification output. This means $P^c_{MD}$ represents the actual probability that a sample belongs the class $c$.

**Uncertainty Prediction.** We define the predictive uncertainty, which is the uncertainty in the network prediction as

$$u_{c^*} = 1 - P^{c^*}_{MD}$$ (6)

For small values of MD, $u_{c^*}$ is around 0 and goes to 1 as the MD increases.

4. Experiments

We compare MAPLE with the following related works: two multi forward-pass methods MC-Dropout [13] (10 dropout samples) and Deep ensemble [26] (10 models), four single forward-pass methods: DUQ [46], SNGP [31], DUE [45] and VMDLS [7]. Following the standard evaluation on OOD detection, we evaluate the methods on classification, predictive probability calibration, and OOD detection on the following benchmark datasets: CIFAR10 [25] vs. SVHN [36]/CIFAR100 [25], CIFAR100 vs. CIFAR10/Tiny ImageNet [28] and ImageNet [40] vs. ImageNet-O [20].

Additionally, we compare the ID metrics for the corrupted version of CIFAR100 [20]. We also compare MAPLE with MD-based methods on OOD detection, namely, the approach by Lee et al. [29], Marginal MD [24] and RMD [39]. We used the near-OOD CIFAR10 vs. CIFAR100 for the comparison, which is notably challenging for OOD detection.

4.1. Evaluation Metrics

We report the standard evaluation metrics [31, 46] namely, the classification accuracy, the Expected Calibration Error (ECE), the Negative Log-Likelihood (NLL), the Area Under the Receiver Operating Characteristics (AUROC) and the Area Under the Precision-Recall curve (AUPR). For qualitative analysis, we use calibration plots. As mentioned previously, MAPLE produces two classification outputs, so we report the accuracies obtained from both the traditional softmax probability and the MD-based classification (Sec. 3.3). The ECE and the NLL are calculated from the predictive probability $P^{c^*}_{MD}$. AUROC and AUPR are calculated from the uncertainty $u_{c^*}$. The definition of these standard metrics are recalled in Appendix A.

4.2. Implementation Details

The CIFAR10 and CIFAR100 training follows [31, 45] and uses a Wide ResNet 28-10 [51] for the classification backbone. The hyperparameters for the trainings are $p = 10, t = 0.3$ and $\text{max\_num\_cluster}= 5$. The ImageNet training is performed on ResNet-50 [18]. The hyperparameters are $p = 20, t = 0.2$ and $\text{max\_num\_cluster}= 5$. Additional details on the dataset splits, hyperparameters, and the hardware used for training are provided in the Appendix B.

4.3. Results

We report the results on CIFAR10, CIFAR100 and ImageNet in Table (Tab.) 1, 2 and 3 respectively.

**OOD Detection Results.** MAPLE outperforms the baseline methods by up to 12% on OOD detection. Note that competitive approaches, such as SNGP and DUE, derive their performance from spectral normalization and Gaussian process layer, which are invasive training add-ons. In contrast, MAPLE relies only on the layers of a standard CNN architecture to achieve superior performance.

**Classification Results.** MAPLE achieves results competitive to state-of-the-art, only 1% below the top method Deep ensemble [26] whose score comes at the cost of
Table 1. CIFAR10 (ID) vs SVHN/CIFAR100 (OOD). Results are averaged over 10 seeds. MAPLE outperforms all single and multi-pass methods on OOD detection, and is significantly faster. Classification with MAPLE is very competitive with the state-of-the-art and the predicted probabilities are better calibrated. Blue: classification based on prediction from softmax probability. Orange: MD-based classification.

Table 2. CIFAR100 (ID) vs CIFAR10/Tiny ImageNet (OOD). Results are averaged over 10 seeds. We also evaluate the ID metrics for the corrupted version of CIFAR100 from [20]. MAPLE achieves the best performance on OOD detection. It is very competitive with other single-pass methods on the classification task. Blue: Classification based on prediction from softmax probability Orange: MD-based classification.

Table 3. ImageNet (ID) vs ImageNet-O (OOD). Results are averaged over 10 seeds. MC-Dropout is performed for 10 forward passes. MAPLE achieves the best performance on OOD detection. It is very competitive with other single-pass methods on the classification task. Blue: Classification based on prediction from softmax probability Orange: MD-based classification.

Training and inference on several models. Note that both MAPLE accuracies, the softmax accuracy and the MD-based one are close. A finer analysis of the accuracy shows that the slight difference in accuracy is in high-confidence predictions (above 80% and 90% confidence) and the accuracy slightly decreases for lower-confidence predictions. See Appendix. D.1 for an extended analysis.

Figure 4. Calibration plots. A perfectly calibrated plot is when the predicted confidence equals the true likelihood i.e., the accuracy. This is shown by the linear dotted line in the plots.

Calibration Results. MAPLE is competitive with state-of-the-art SNGP [31] and Deep Ensembles. The calibration plot for CIFAR10 and CIFAR100 and shown in Fig. 4. On CIFAR10, all methods are well-calibrated, except for MC-Dropout that is overconfident in its predictions, which explains its high ECE score. When the accuracy is below 0.4, baseline methods become overconfident, whereas MAPLE is closer to optimal calibration and achieves the best ECE score.

Additional results on Gaussian test, are provided in Appendix. D.

4.4. Comparison with other MD methods

Setup. MAPLE is compared against MD-based OOD detectors [24, 29, 39]. These methods are tailored for OOD detection, so we report the metric relevant to this task only for the sake of fairness. We report the AUROC score on the challenging near-OOD dataset CIFAR10 vs. CIFAR100. The experiments are done with a Wide ResNet 28-10 [51].

Table 4. Comparison with MD-based OOD detection. MAPLE performs better in OOD detection than existing MD-based methods on the CIFAR10 vs. CIFAR100 setup.

OOD Detection Results. MAPLE achieves top-performance on Near-OOD detection (Tab. 4), which supports MAPLE’s representation regularization. Note that the primary difference between MAPLE and the baselines is their lack of constraints on the latent representation. In contrast, we force the samples of every class to be Gaussian be-
To study the influence of triplet loss, we train using the cross-entropy loss and clustering. We compare Mahalanobis distance values. The error is more pronounced when the samples deviate from the Gaussian distribution by a large factor. This explains why the MD-based approaches underperform compared to MAPLE on Near-OOD.

### 4.5. Ablation analysis

In this study, we assess how the different components of MAPLE impact its performance. We train a Wide ResNet 28-10 [51] network on CIFAR10 and use SVHN and CIFAR100 as OOD datasets.

#### Dimensionality Reduction

We consider two scenarios:
1. **DNN+MD** - A baseline where a standard Deep Neural Network (DNN) is trained with the cross-entropy loss and with no feature regularization. The MD is computed on the raw features, and we add a value of $1e^{-20}$ to the diagonal elements [39] to avoid a singular covariance matrix.
2. **DNN+PCA+MD** - It follows (1) except that the MD is derived on PCA-reduced features.

**Results:** Dimensionality reduction (2) drastically improves the network’s performance, as shown in the first line of Tab. 5. The improvement amounts to 7-30% on the OOD metrics and 3% on the ID metrics. One possible explanation is that the reduced dimensions are the ones that contribute to distinguishing ID samples from OOD ones, as previously observed by [39]. When including all the feature dimensions in the MD, the dimensions that do not contribute to discriminating ID and OOD samples add up and dominate the final MD score.

#### Distance Definition

We compare Mahalanobis distance and Euclidean distance (ED) in the network’s latent space. We compare (2) **DNN+PCA+MD** with the new experiment (3) **DNN+PCA+ED** - It follows (2) except that the MD is replaced with ED. As for MD, the $\chi^2_d$ distribution is used to obtain the probability values from ED (Sec 3.3).

**Results:** The results show that MD boosts the performance in terms of ID and OOD metrics. The improvement is ECE score by 5%, and the OOD metrics improved by 3-9% when using MD. This is because MD takes into account the data correlation, which gives a better estimate of the probability and uncertainty values.

#### Representation training

To study the influence of the training on the representations, we consider three experiments:
- (4) **DNN+Triplet+PCA+MD** - We train the DNN using both cross-entropy and triplet loss.
- (5) **DNN+Clustering+PCA+MD** - We train using the cross-entropy loss only and periodically cluster the feature points using X-Means.
- (6) **MAPLE** - This is our proposed method that fuses (4) and (5). For all experiments, the MD is derived on the reduced features.

**Results:** Using the triplet loss (4) improves the performance considerably compared to training with the cross-entropy loss only (2). An explanation is that the triplet loss pulls in-class feature embeddings together, and pushes the other class features apart. This encourages the representations to be well separated and makes it easier to distinguish OOD features. Choosing the triplet loss for metric learning is empirically motivated: experiments using contrastive loss showed that triplet loss has a slightly better performance.

Periodic clustering (5) improves the ECE score by 2%, and the AUROC and AUPR scores on SVHN by about 7% compared to (2). However, there is a slight drop in accuracy by 0.3% and OOD metric by 4% on CIFAR100. One explanation is that clustering increases the chances of new classes to overlap. This phenomenon is illustrated in the centre plot of Fig. 3. The class overlap is particularly hindering when the new domain is close to the training one: with clustering (5), the SHVN scores are better but the near-OOD CIFAR100 performs better without clustering (2).

MAPLE uses clustering together with triplet loss and achieves top-performance. The triplet loss reduces the overlap introduced with the clustering by pulling apart the newly created classes. With MAPLE, the latent representations are approximately Gaussian and well-clustered resulting in better MD estimates and superior performance in both ID and OOD metrics. Compared to experiment (2), the calibration error drops by 4% and the OOD scores improved by 4-11%.

<table>
<thead>
<tr>
<th>Method</th>
<th>Softmax Accuracy ↑</th>
<th>ID-based Accuracy ↑</th>
<th>ECE ↓</th>
<th>OOD metrics - SVHN</th>
<th>OOD metrics—CIFAR100</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNN+MD (1)</td>
<td>0.990</td>
<td>0.943</td>
<td>0.086</td>
<td>0.752</td>
<td>0.583</td>
</tr>
<tr>
<td>DNN+PCA+MD (2)</td>
<td>0.990</td>
<td>0.946</td>
<td>0.053</td>
<td>0.855</td>
<td>0.813</td>
</tr>
<tr>
<td>DNN+PCA+ED (3)</td>
<td>0.930</td>
<td>0.943</td>
<td>0.105</td>
<td>0.829</td>
<td>0.734</td>
</tr>
<tr>
<td>DNN+Triplet+PCA+MD (4)</td>
<td>0.947</td>
<td>0.945</td>
<td>0.032</td>
<td>0.922</td>
<td>0.811</td>
</tr>
<tr>
<td>DNN+Clustering+PCA+MD (5)</td>
<td>0.947</td>
<td>0.945</td>
<td>0.032</td>
<td>0.922</td>
<td>0.811</td>
</tr>
<tr>
<td>MAPLE (6)</td>
<td>0.996</td>
<td>0.994</td>
<td>0.012</td>
<td>0.996</td>
<td>0.930</td>
</tr>
</tbody>
</table>

Table 5. Ablation study. We evaluate the influence of several MAPLE components. PCA (1 vs 2) results in a significant improvement of the OOD detection by discarding non-informative dimensions. The distances derived on these reduced features are better representative of the similarity between the input samples. The MD (2 vs 3) is better suited than ED for calibrated classification and OOD detection, which reiterates conclusions already found in previous works. The triplet loss (2 vs 4) improves both the accuracy and the OOD metrics by increasing the class separation. Clustering alone (2 vs 5) also contributes to a better separation of the classes, but the results are not as significant. The joint use of triplet loss and clustering, as done in MAPLE (6) achieves the best results on both classification and OOD detection. Note: #Eig refers to the number of principal components, whenever applicable.
False Negative Ratio $t$. We evaluate the influence of the clustering trigger i.e., the False negatives ratio. We train MAPLE with a range of $t$ values on CIFAR10 (Tab. 6).

Results: A low value of $t$ results in over-clustering, where multiple clusters contain similar images. This further increases the chances of misclassifications, leading to decrease in the metric values. On the other hand, high $t$ values result in under-clustering. Note that for $t > 0.3$, there are no additional clusters generated. This is because, the classes have false negative ratios that are below this threshold and so, they are not clustered. For CIFAR10, a $t$ value of $0.3$ yields the best results.

An extended ablation analysis on the influence of classification backbones, clustering methods, and hyperparameters is provided in Appendix. E.

5. Discussion

With the periodical clustering and the dynamic re-labeling, a natural question that arises is *Is there a drop in performance when the ground truth labels change during training?*. Experimentally, we observe a drop in training accuracy by 2-3% in the following epoch after every clustering phase. However, the network makes up for the drop within 4-5 epochs of training.

It can happen that the clusters contain very few samples, which introduces label imbalance when classifying. This is exacerbated when the samples are over-clustered. To mitigate this, we restrict X-Means to only cluster the classes that get misclassified. These are the classes with a false negative ratio higher than the threshold $t$. Automatic clustering regularization [4, 22, 27] is left for future work.

6. Use Case: Microorganism Classification

We consider the real-life computer vision use-case of image-based diatom identification [8]. Diatoms are microorganisms present in the water. The distribution of diatoms in the water is a useful indicator for predicting the water quality. Diatoms consist of several species or 'taxa', each corresponding to a different class with a different appearance. Typical in several biology applications, the image dataset includes a lot of intra-class variance (Fig. 5). In this study, we evaluate the performance of different approaches when encountering taxa that were not previously trained on.

We train a Wide ResNet 28-10 on 130 taxa and use 36 taxa as OOD. The dataset is particularly challenging since it is fine-grained and Near-OOD. Additional details on the dataset and experimental setup are provided in Appendix B.4. As shown in Tab. 7, MAPLE outperforms all baselines on OOD detection. While Deep Ensemble has a slightly better classification accuracy and ECE score, MAPLE significantly outperforms it in OOD with a 30% score boost and a runtime 8 times faster.

Figure 5. *Micro-organisms belonging to the same class.* These images of one diatom class show wide appearance changes due to different viewpoints during the acquisition. These translate into separate distributions in the latent space, deviating from Gaussian distribution. MAPLE’s regularization makes the latent space Gaussian, hence suitable for MD calculation.

7. Conclusion

This paper presents MAPLE, a self-supervised regularization method for uncertainty estimation and out-of-distribution detection on CNN classifiers. The uncertainty is derived from the Mahalanobis Distance (MD) between an image representation and the class representations in the network’s latent space. MAPLE derives meaningful MD distances by introducing a regularizer based on self-supervised label refinement and metric learning. Thus, MAPLE learns well-clustered representations that are approximately Gaussian for each class, which complies with the theoretical requirements of MD-based uncertainty estimation. Experimental results show that MAPLE achieves state-of-the-art results on out-of-distribution detection and is very competitive with existing methods on predictive probability calibration. MAPLE also has the significant advantage of introducing the least architectural changes. Finally, we demonstrate a real-life use-case of our method on microorganism classification for the automatic assessment of water quality in natural ecosystems.
References


