Robust Variational Autoencoding with Wasserstein Penalty for Novelty Detection

Chieh-Hsin Lai[*]
School of Mathematics,
University of Minnesota

Dongmian Zou[*]
Division of Natural and Applied Sciences,
Duke Kunshan University

Gilad Lerman
School of Mathematics,
University of Minnesota

Abstract

We propose a new method for novelty detection that can tolerate high corruption of the training points, whereas previous works assumed either no or very low corruption. Our method trains a robust variational autoencoder (VAE), which aims to generate a model for the uncorrupted training points. To gain robustness to high corruption, we incorporate the following four changes to the common VAE: 1. Extracting crucial features of the latent code by a carefully designed dimension reduction component for distributions; 2. Modeling the latent distribution as a mixture of Gaussian low-rank inliers and full-rank outliers, where the testing only uses the inlier model; 3. Applying the Wasserstein-1 metric for regularization, instead of the Kullback-Leibler (KL) divergence; and 4. Using a robust error for reconstruction. We establish both robustness to outliers and suitability to low-rank modeling of the Wasserstein metric as opposed to the KL divergence. We illustrate state-of-the-art results on standard benchmarks.

1 INTRODUCTION

Machine learning solutions often assume that training datasets are flawless and can serve as ground truth. However, this assumption usually does not hold in practice. Indeed, most datasets, even commonly used ones such as CIFAR-10 or ImageNet, suffer from corruption and mislabeling (Northcutt et al., 2021). While in many applications the percentage of mislabels may be sufficiently small, there are important scenarios where this is not the case. One such scenario appears when studying problems with no earlier experience and expertise. For instance, in the beginning of the COVID-19 pandemic it was hard to diagnose COVID-19 patients and distinguish them from other patients with pneumonia (Chowdhury et al., 2020; Xiao et al., 2020). Another scenario occurs when it is very hard to make precise measurements, for example, when working with the highly corrupted images in cryogenic electron microscopy (cryo-EM) (Miolane et al., 2020; Huang & Tagare, 2015).

One problem, where it is crucial to carefully address mislabeled training data points, is novelty detection. It asks to detect testing data points that deviate from the underlying structure of a given training dataset (Chandola et al., 2009; Pimentel et al., 2014; Chalapathy & Chawla, 2019; Perera et al., 2021). Novelty detection is equivalent to the well-known one-class classification problem (Moya & Hush, 1996). This problem asks to identify members of a class in a test dataset, and consequently distinguish them from "novel" data points, given training points from this class. The points of the main class are commonly referred to as inliers and the novel ones as outliers. Novelty detection is also commonly referred to as semi-supervised anomaly detection. In this terminology, the notion of being "semi-supervised" is different from usual, and means that a training set is provided for the inliers only. On the other hand, the supervised case has labeled training data for both the inliers and outliers, and the unsupervised case has no training and is also known as "outlier detection".

Traditional one-class classification methods often assume that the training set is purely sampled from a single class or has few outliers and perform poorly when there is a non-trivial portion of outliers. In this paper, we study a robust version of novelty detection that allows a nontrivial fraction of corrupted samples, namely outliers, within the training set. We solve this problem by using a special variational autoencoder (VAE) (Kingma & Welling, 2014). Our VAE is able to model the underlying distribution of the uncorrupted data, despite nontrivial corruption. We refer to it as "Mixture Autoencoding with Wasserstein penalty", or "MAW".

[*] indicates equal contribution.

Proceedings of the 26[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

1.1 Previous Work

Solutions to novelty detection either estimate the density of the inlier distribution (Bengio & Monperrus, 2005; Ilonen et al., 2006) or determine a geometric property of the inliers, such as their boundary set (Breunig et al., 2000; Schölkopf et al., 2000; Xiao et al., 2016; Wang & Lan, 2020; Jiang et al., 2019). When the inlier distribution is nicely approximated by a low-dimensional linear subspace, Shyu et al. (2003) propose to distinguish between inliers and outliers via Principal Component Analysis (PCA). In order to consider more general cases of nonlinear low-dimensional structures, one may use autoencoders (or restricted Boltzmann machines), which nonlinearly generalize PCA (Goodfellow et al., 2016, Ch. 2) and whose reconstruction error naturally provides a score for membership in the inlier class. Instances of this strategy with various architectures include (Zhai et al., 2016; Zong et al., 2018; Sabokrou et al., 2018; Perera et al., 2019; Pidhorskyi et al., 2018). In all of these works, but (Zong et al., 2018), the training set is assumed to solely represent the inlier class. If there are also outliers (with a simple shape) among the inliers (with a complex shape), encoding the inlier distribution becomes difficult. Nevertheless, some previous works already explored the possibility of a corrupted training set (Xiao et al., 2016; Wang & Lan, 2020; Zong et al., 2018). In particular, Xiao et al. (2016); Zong et al. (2018) test artificial instances with at most 5% corruption of the training set and Wang & Lan (2020) consider ratios of 10%, but with very small numbers of training points. In this work we consider corruption ratios up to 50%, with a method that tries to estimate the distribution of the training set, and not just a geometric property.

VAEs (Kingma & Welling, 2014) have been commonly used for generating distributions with reconstruction scores and are thus natural for novelty detection without corruption. The first VAE-based method for novelty detection was suggested by An & Cho (2015). It was recently extended by Daniel et al. (2021) who modified the training objective. A variety of VAE models were also proposed for special anomaly detection problems, which are different from novelty detection (Xu et al., 2018; Zhang et al., 2019; Pol et al., 2019). Current VAE-based methods for novelty detection do not perform well when the training data is corrupted. Indeed, the learned distribution of any such method also represents corruption, that is, the outlier component. To the best of our knowledge, no effective solutions were proposed for collapsing the outlier mode so that the trained VAE would only represent the inlier distribution.

A variant of VAE is the adversarial autoencoder (AAE) of (Makhzani et al., 2016). The penalty term of AAE takes the form of a generative adversarial network (GAN) (Goodfellow et al., 2016), where the AAE's encoder serves as the GAN's generator. We can thus view it as a hybrid GAN-VAE model. Another such model is the Wasserstein autoencoder (WAE) (Tolstikhin et al., 2018), which generalizes AAE by allowing a general objective function. Our proposed model is also a hybrid GAN-VAE. Other hybrid VAE-GAN models include (Mescheder et al., 2017; Xian et al., 2019; Ye & Bors, 2021). The GAN of (Mescheder et al., 2017) is used for both the samples and the latent code, the GAN of (Xian et al., 2019; Ye & Bors, 2021) is used only for the samples, whereas the GAN of our work and (Makhzani et al., 2016; Tolstikhin et al., 2018) is used only for the latent code. We demonstrate the resulting robustness to outliers due to our particular use of a GAN.

There are two relevant lines of work on robustness to outliers in linear modeling that can be used in nonlinear settings via autoencoders or VAEs. Robust PCA aims to deal with sparse elementwise corruption of a data matrix (Candès et al., 2011; De La Torre & Black, 2003; Wright et al., 2009; Vaswani & Narayanamurthy, 2018). Robust subspace recovery (RSR) aims to address general corruption of selected data points and thus better fits the framework of outliers (Watson, 2001; De La Torre & Black, 2003; Ding et al., 2006; Zhang et al., 2009; McCoy & Tropp, 2011; Xu et al., 2012; Lerman & Zhang, 2014; Zhang & Lerman, 2014; Lerman et al., 2015; Lerman & Maunu, 2017; Maunu et al., 2019; Lerman & Maunu, 2018; Maunu & Lerman, 2019). Autoencoders that use robust PCA for anomaly detection tasks were proposed in (Chalapathy et al., 2017; Zhou & Paffenroth, 2017). It is shown in (Dai et al., 2018) that a VAE can be interpreted as a nonlinear robust PCA problem. Nevertheless, explicit regularization is often required to improve robustness to sparse corruption in VAEs (Akrami et al., 2019; Eduardo et al., 2020). An RSR layer was successfully applied to outlier detection in (Lai et al., 2020). One can also apply this work to novelty detection.

We remark that the setting of our work is different from that of out-of-distribution (OOD) detection and open-set recognition. Indeed, in these recent settings the inliers are from multiple classes that need to be identified. On the other hand, this work does not ask to classify the inliers.

1.2 This Work

We propose a robust novelty detection procedure, MAW, that aims to model the distribution of the training data in the presence of a nontrivial fraction of outliers. We highlight its following four features:

1. MAW models the latent distribution by a Gaussian mixture of low-rank inliers and full-rank outliers, and applies the inlier distribution for testing. Previous applications of mixture models for novelty detection were designed for multiple modes of inliers and used more complicated tools such as additional network construction (Zong et al., 2018) or clustering (Aytekin et al., 2018; Lee et al., 2018).

2. MAW applies a novel dimension reduction component, which extracts lower-dimensional features of the latent distribution. The reduced dimension allows using full covariances; whereas previous VAE-based methods for novelty detection used diagonal covariances in their models (An & Cho, 2015; Daniel et al., 2021).
3. MAW uses the Wasserstein-1 (W_1) metric for the latent code penalty. We prove that the Wasserstein metric gives rise to outlier-robust estimation and is suitable to the low-rank modeling of inliers by MAW. We also show that these properties do not hold for the commonly-used KL divergence. To the best of our knowledge, this is the first theoretical analysis that clarifies the advantage of the Wasserstein distance over the KL divergence in a VAE in terms of robustness to outliers and low-rank inlier modeling.
4. MAW achieves state-of-the-art results on popular anomaly detection datasets.

Additional two features are as follows. First, for reconstruction, MAW replaces the common least squares formulation with a least absolute deviations formulation. This can be justified by the use of a robust estimator (Lopuhaa & Rousseeuw, 1991) with a heavier-tail likelihood. Second, MAW is attractive for practitioners. It is simple to implement in any standard deep learning library, and is easily adaptable to other choices of network architecture, energy functions and similarity scores.

2 DESCRIPTION OF MAW

We motivate and overview the underlying model and assumptions of MAW in §2.1. We describe the implementation details of its components in §2.2 and sketch the algorithm procedures in the supplementary materials. Fig. 1 illustrates the general idea of MAW and can assist in reading this section.

2.1 The Model and Assumptions of MAW

MAW aims to robustly estimate a mixture inlier-outlier distribution for the training data and then use its inlier component to detect outliers in the testing data. For this purpose, it designs a novel variational autoencoder with an underlying mixture model and a robust loss function in the latent space. We find the variational framework natural for novelty detection. Indeed, it learns a distribution that describes the inlier training examples and generalizes to the inlier test data. Moreover, the variational formulation allows a direct modeling of a Gaussian mixture model (GMM) in the latent space, unlike a standard autoencoder.

Let $\mathbf{x}$ be a random variable in $\mathbb{R}^D$ with an unknown training data distribution, which contains both inlier and outlier modes. We assume L training points in $\mathbb{R}^D$, $\{\mathbf{x}^{(i)}\}_{i=1}^L$ sampled from this distribution. We assume a latent random variable $\mathbf{z}$ of low and even dimension $2 \leq d \leq D$ (our default choice is $d = 2$), and a standardized Gaussian prior, $p(\mathbf{z})$, so that $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{d\times d})$. In the remaining text, we shall denote it as

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \boldsymbol{I}_{d\times d}).$$

The posterior distribution $p(\mathbf{z}|\mathbf{x})$ is unknown. However, we assume an approximation to it, which we denote by

$$q(\mathbf{z}|\mathbf{x}) = \eta\mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + (1-\eta)\mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2), \quad (1)$$

where $\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2$ depend on $\mathbf{x}$ and are generated by the encoder network and the dimension reduction component (explained below) the default choice for the mixture parameter is $\eta = 5/6$ (the low sensitivity of our method to the choice of η is demonstrated in the supplementary materials). The first mode in (1) represents the inliers and the second one represents the outliers. We model $p(\mathbf{z})$ as a single Gaussian since we only want to include the inlier information, while having the simplest possible design. In §4.3, we will numerically compare with the modeling of $p(\mathbf{z})$ as a GMM. In the supplementary materials we intuitively clarify the mechanism that helps in such modeling.

The dimension reduction component involves a mapping from a higher-dimensional space onto the latent space. It is analogous to the RSR layer (Lai et al., 2020) that projects encoded points onto the latent space, but requires a more careful design since we consider a distribution rather than sample points. Due to this reduction, we assume that the mapped covariance matrices of $\mathbf{z}|\mathbf{x}$ are full, unlike common single-mode VAE models that assume a diagonal covariance (Kingma & Welling, 2014; An & Cho, 2015). We assume that the inliers lie on a low-dimensional structure and we thus enforce the lower rank $d/2$ for $\boldsymbol{\Sigma}_1$, but allow $\boldsymbol{\Sigma}_2$ to have full rank d. Nevertheless, we later describe a necessary regularization of both matrices. We remark that the low rank assumption results in the main distinction between the inliers and outliers in (1) (as noted in the supplementary materials the choice of $\eta > 0.5$ is not crucial).

The unknown posterior distribution $p(\mathbf{z}|\mathbf{x})$ is approximated within the variational family $\mathcal{Q} = \{q(\mathbf{z}|\mathbf{x})\}$ indexed by $\boldsymbol{\mu}_1$, $\boldsymbol{\Sigma}_1$, $\boldsymbol{\mu}_2$ and $\boldsymbol{\Sigma}_2$. Unlike a standard VAE, which maximizes the evidence lower bound (ELBO), MAW maximizes the following loss function, which uses the W_1 distance (defined in the supplementary materials), instead of the KL divergence, for regularizing the log-likelihood of the data distribution:

$$\mathcal{L}_{\text{MAW}}(q) = \mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z}) - W_1(q(\mathbf{z}), p(\mathbf{z})) \,. \quad (2)$$

We use the Wasserstein distance since it is more robust to outliers than the KL divergence and is thus more suitable for detecting anomalies (see related guarantees in §3).

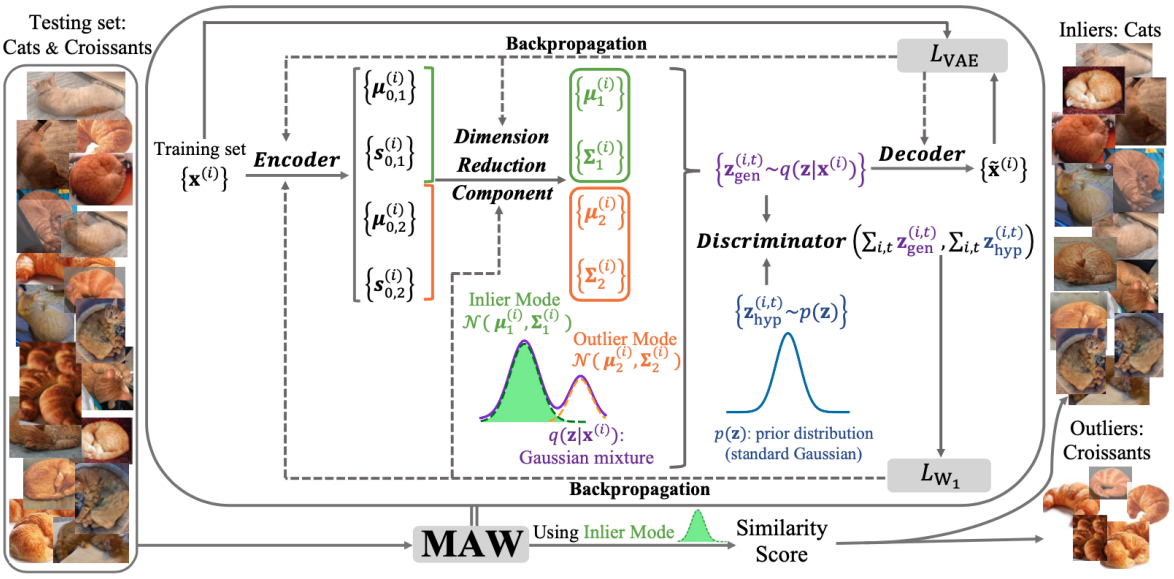

Figure 1: Demonstration of the architecture of MAW for novelty detection.

Following the VAE framework, we use a Monte-Carlo approximation to estimate $\mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z})$ with i.i.d. samples, $\{\mathbf{z}^{(t)}\}_{t=1}^{T}$, from $q(\mathbf{z}|\mathbf{x})$ as follows:

$$\mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z}) \approx \frac{1}{T} \sum_{t=1}^{T} \log p(\mathbf{x}|\mathbf{z}^{(t)}). \tag{3}$$

To enhance robustness, we let the negative log likelihood function $-\log p(\mathbf{x}|\mathbf{z}^{(t)})$ be proportional to the ℓ_2 norm of the difference of the random variable $\mathbf{x}$ and a mapping of the sample $\mathbf{z}^{(t)}$ from $\mathbb{R}^d$ to $\mathbb{R}^D$ by the decoder, $\mathcal{D}$, that is,

$$-\log p(\mathbf{x}|\mathbf{z}^{(t)}) \propto \left\| \mathbf{x} - \mathcal{D}(\mathbf{z}^{(t)}) \right\|_2 . \tag{4}$$

We deviate from the common choice of the squared ℓ_2 norm, which corresponds to an underlying Gaussian likelihood and assume instead a likelihood with a heavier tail.

MAW trains its networks by minimizing $-\mathcal{L}_{\text{MAW}}(q)$. For $1 \le i \le L$, it samples $\{\mathbf{z}_{\text{gen}}^{(i,t)}\}_{t=1}^{T}$ from $q(\mathbf{z}|\mathbf{x}^{(i)})$, where all samples are independent. Using the aggregation formula $q(\mathbf{z}) = L^{-1} \sum_{i=1}^{L} q(\mathbf{z}|\mathbf{x}^{(i)})$, the approximation of $p(\mathbf{x})$ by the empirical distribution of the training data, and (2)-(4), MAW applies the following approximation of $-\mathcal{L}_{\text{MAW}}(q)$:

$$\begin{aligned} &\frac{1}{LT} \sum_{i=1}^{L} \sum_{t=1}^{T} \left\| \mathbf{x}^{(i)} - \mathcal{D}(\mathbf{z}_{\text{gen}}^{(i,t)}) \right\|_2 \\ &\qquad + W_1 \left(\frac{1}{L} \sum_{i=1}^{L} q(\mathbf{z}|\mathbf{x}^{(i)}), p(\mathbf{z}) \right). \end{aligned} \tag{5}$$

Our procedure of minimizing (5) is described in §2.2.

During testing, MAW identifies outliers according to low similarity scores computed between test points and points generated from the learned inlier component of $\mathbf{z}|\mathbf{x}$.

2.2 Details of Implementing MAW

MAW has a VAE-type structure with additional WGAN-type structure for minimizing the W_1 loss in (5). We provide here details of implementing these structures. Some specific choices of the networks are described in §4 since they may depend on the type of datasets.

The VAE-type structure of MAW contains three ingredients: encoder, dimension reduction component and decoder. The encoder forms a neural network (NN), $\mathcal{E}$, that maps the training sample $\mathbf{x}^{(i)}$ in $\mathbb{R}^D$ to $\boldsymbol{\mu}_{0,1}^{(i)}, \boldsymbol{\mu}_{0,2}^{(i)}, \boldsymbol{s}_{0,1}^{(i)}, \boldsymbol{s}_{0,2}^{(i)}$ in $\mathbb{R}^{D'}$, where our default choice is $D' = 128$. The dimension reduction component then computes the following statistical quantities of the GMM $\mathbf{z}|\mathbf{x}^{(i)}$: means $\boldsymbol{\mu}_1^{(i)}$ and $\boldsymbol{\mu}_2^{(i)}$ in $\mathbb{R}^d$ and covariance matrices $\boldsymbol{\Sigma}_1^{(i)}$ and $\boldsymbol{\Sigma}_2^{(i)}$ in $\mathbb{R}^{d \times d}$. First, a linear layer, represented by $\boldsymbol{A} \in \mathbb{R}^{D' \times d}$, maps (via $\boldsymbol{A}^{\mathrm{T}}$) the features $\boldsymbol{\mu}_{0,1}^{(i)}, \boldsymbol{\mu}_{0,2}^{(i)} \in \mathbb{R}^{D'}$ to the following respective vectors in $\mathbb{R}^d$:

$$\boldsymbol{\mu}_1^{(i)} = \boldsymbol{A}^{\mathrm{T}} \boldsymbol{\mu}_{0,1}^{(i)} \text{ and } \boldsymbol{\mu}_2^{(i)} = \boldsymbol{A}^{\mathrm{T}} \boldsymbol{\mu}_{0,2}^{(i)}.$$

The mapping of the covariance matrices is constructed as follows. Form $\boldsymbol{M}_j^{(i)} = \boldsymbol{A}^{\mathrm{T}} \mathrm{diag}(\boldsymbol{s}_{0,j}^{(i)}) \boldsymbol{A}$ for $j = 1, 2$. For $j = 2$, compute $\boldsymbol{\Sigma}_2^{(i)} = \boldsymbol{M}_2^{(i)} \boldsymbol{M}_2^{(i)\mathrm{T}}$. For $j = 1$, we first need to reduce the rank of $\boldsymbol{M}_1^{(i)}$. For this purpose, we form

$$\boldsymbol{M}_1^{(i)} = \boldsymbol{U}_1^{(i)} \mathrm{diag}(\boldsymbol{\sigma}_1^{(i)}) \boldsymbol{U}_1^{(i)\mathrm{T}}, \tag{6}$$

the spectral decomposition of $\boldsymbol{M}_1^{(i)}$, and then truncate its bottom $d/2$ eigenvalues. That is, let $\tilde{\boldsymbol{\sigma}}_1^{(i)} \in \mathbb{R}^d$ have the same entries as the largest $d/2$ entries of $\boldsymbol{\sigma}_1^{(i)}$ and zero en-

tries otherwise. Then, compute

$$\tilde{\boldsymbol{M}}_1^{(i)} = \boldsymbol{U}_1^{(i)} \mathrm{diag}(\tilde{\boldsymbol{\sigma}}_1^{(i)}) \boldsymbol{U}_1^{(i)\mathrm{T}} \tag{7}$$

and

$$\boldsymbol{\Sigma}_1^{(i)} = \tilde{\boldsymbol{M}}_1^{(i)} \tilde{\boldsymbol{M}}_1^{(i)\mathrm{T}}.$$

To ensure numerically-significant positive definiteness of both $\boldsymbol{\Sigma}_1^{(i)}$ and $\boldsymbol{\Sigma}_2^{(i)}$, we add to them an identity matrix. Despite this, the low-rank structure of $\boldsymbol{\Sigma}_1^{(i)}$ is still evident. Note that the dimension reduction component only trains $\boldsymbol{A}$. The decoder, $\mathcal{D} : \mathbb{R}^d \to \mathbb{R}^D$, maps independent samples, $\{\mathbf{z}_{\text{gen}}^{(i,t)}\}_{t=1}^T$, generated for each $1 \le i \le L$ by the distribution

$$\eta \mathcal{N}(\boldsymbol{\mu}_1^{(i)}, \boldsymbol{\Sigma}_1^{(i)}) + (1-\eta)\mathcal{N}(\boldsymbol{\mu}_2^{(i)}, \boldsymbol{\Sigma}_2^{(i)}),$$

into the reconstructed data space.

The loss function associated with the VAE structure is the first term in (5). We can write it as

$$L_{\text{VAE}}(\mathcal{E}, \boldsymbol{A}, \mathcal{D}) = \frac{1}{LT} \sum_{i=1}^{L} \sum_{t=1}^{T} \left\| \mathbf{x}^{(i)} - \mathcal{D}(\mathbf{z}_{\text{gen}}^{(i,t)}) \right\|_2 . \tag{8}$$

The dependence of this loss on $\mathcal{E}$ and $\boldsymbol{A}$ is implicit, but follows from the fact that the parameters of the sampling distribution of each $\mathbf{z}_{\text{gen}}^{(i,t)}$ were obtained by $\mathcal{E}$ and $\boldsymbol{A}$.

The WGAN-type structure seeks to minimize the second term in (5) using the dual formulation

$$W_1 \left(\frac{1}{L} \sum_{i=1}^{L} q(\mathbf{z}|\mathbf{x}^{(i)}), p(\mathbf{z}) \right) = \tag{9}$$

$$\sup_{\|f\|_{\text{Lip}} \le 1} \mathbb{E}_{\mathbf{z}_{\text{hyp}} \sim p(\mathbf{z})} f(\mathbf{z}_{\text{hyp}}) - \mathbb{E}_{\mathbf{z}_{\text{gen}} \sim \frac{1}{L}\sum_{i=1}^{L} q(\mathbf{z}|\mathbf{x}^{(i)})} f(\mathbf{z}_{\text{gen}}).$$

The generator of this WGAN-type structure is composed of the encoder $\mathcal{E}$ and the dimension reduction component, which we represent by $\boldsymbol{A}$. It generates the samples $\{\mathbf{z}_{\text{gen}}^{(i,t)}\}_{i=1,t=1}^{L,T}$ described above. The discriminator, $\mathcal{D}is$, of the WGAN-type structure plays the role of the Lipschitz function f in (9). It compares the latter samples with the i.i.d. samples $\{\mathbf{z}_{\text{hyp}}^{(i,t)}\}_{t=1}^T$ from the prior distribution. In order to make $\mathcal{D}is$ Lipschitz, its weights are clipped to $[-1, 1]$ during training. In the MinMax game of this WGAN-type structure, the discriminator minimizes and the generator ($\mathcal{E}$ and $\boldsymbol{A}$) maximizes

$$L_{W_1}(\mathcal{D}is) = \frac{1}{LT} \sum_{i=1}^{L} \sum_{t=1}^{T} \left(\mathcal{D}is(\mathbf{z}_{\text{gen}}^{(i,t)}) - \mathcal{D}is(\mathbf{z}_{\text{hyp}}^{(i,t)}) \right) . \tag{10}$$

We note that maximization of (10) by the generator is equivalent to minimization of the loss function

$$L_{\text{GEN}}(\mathcal{E}, \boldsymbol{A}) = -\frac{1}{LT} \sum_{i=1}^{L} \sum_{t=1}^{T} \mathcal{D}is(\mathbf{z}_{\text{gen}}^{(i,t)}) . \tag{11}$$

During training, MAW alternatively minimizes the losses (8), (10) and (11) instead of their weighted sum. Therefore, any multiplicative constant in front of either term of (5) will not affect the optimization. In particular, it was okay to omit the multiplicative constant of (4) when deriving (5).

For each testing point $\mathbf{y}^{(j)}$, we sample $\{\mathbf{z}_{\text{in}}^{(j,t)}\}_{t=1}^T$ from the inlier mode of the learned latent Gaussian mixture and decode them as $\{\tilde{\mathbf{y}}^{(j,t)}\}_{t=1}^T = \{\mathcal{D}(\mathbf{z}_{\text{in}}^{(j,t)})\}_{t=1}^T$. Using a similarity measure $S(\cdot,\cdot)$ (our default is the cosine similarity), we compute

$$S^{(j)} = \sum_{t=1}^{T} S(\mathbf{y}^{(j)}, \tilde{\mathbf{y}}^{(j,t)}).$$

If $S^{(j)}$ is larger than a chosen threshold, then $\mathbf{y}^{(j)}$ is classified as normal, and otherwise, novel. Additional details of MAW are in the supplementary materials.

We remark that in our setting we find it natural to implement an auxiliary WGAN on top of the VAE component in order to estimate the W_1 distance. We did not find it useful to directly estimate the W_1 distance by either the sliced Wasserstein distance (Kolouri et al., 2018, 2019) or the Sinkhorn algorithm (Cuturi, 2013). Indeed, it is not clear how to use these methods in order to minimize the estimated W_1 distance with respect to the parameters within the neural network. In particular, the partial derivatives for learning the GMM using the sliced Wasserstein distance already have very complicated forms, and it is very difficult to include them in our framework, where neural networks are involved.

3 THEORETICAL GUARANTEES

We theoretically establish the superiority of using the Wasserstein distance over the KL divergence, where we leave out some details (in particular proofs) to the supplementary materials. We formulate a mathematical setting that aims to isolate the minimization of the WGAN-type structure introduced in §2.2, while ignoring unnecessary complex components of MAW. We assume a mixture parameter $\eta > 1/2$, a separation parameter $\epsilon > 0$ and denote by $\mathcal{R}$ the regularizing function, which can be either the KL divergence or the Wasserstein distance, and by $\mathcal{S}_+^K$ and $\mathcal{S}_{++}^K$ the sets of $K \times K$ positive semidefinite and positive definite matrices, respectively. Our mathematical setting, which we motivate in the supplementary materials, assumes $\boldsymbol{\mu}_0 \in \mathbb{R}^K$ and $\boldsymbol{\Sigma}_0 \in \mathcal{S}_{++}^K$ and requires to minimize

$$\min_{\substack{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathbb{R}^K; \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2 \in \mathcal{S}_+^K \\ \text{s.t. } \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 \ge \epsilon}} \eta \mathcal{R}\left(\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)\right) \tag{12}$$

$$+ (1-\eta)\mathcal{R}\left(\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2), \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)\right).$$

This minimization aims to approximate the "prior" distri-

bution $\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ with a Gaussian mixture distribution. For MAW, $\boldsymbol{\mu}_0 = \mathbf{0}$ and $\boldsymbol{\Sigma}_0 = \boldsymbol{I}$, but our generalization helps clarify things. The constraint $\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 \geq \epsilon$ distinguishes between the inlier and outlier modes and it is a realistic assumption as long as ϵ is sufficiently small.

3.1 Guarantees for (12) with Identical Covariances

Our cleanest result is when $\boldsymbol{\Sigma}_0$, $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ coincide. It is formulated next and demonstrates robustness to the outlier component by the W_1 (or W_p, $p \geq 1$) minimization and not by the KL minimization (its proof is in the supplementary materials).

Proposition 3.1 *If $\boldsymbol{\mu}_0 \in \mathbb{R}^K$, $\boldsymbol{\Sigma}_0 \in \mathcal{S}_{++}^K$, $\epsilon > 0$ and $1 > \eta > 1/2$, then the minimizer of (12) with $\mathcal{R} = W_p$, $p \geq 1$ and the additional constraint: $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2$, satisfies $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_0$, and thus the recovered inlier distribution coincides with the "prior distribution". However, the minimizer of (12) with $\mathcal{R} = KL$ and the same constraint satisfies $\boldsymbol{\mu}_0 = \eta\boldsymbol{\mu}_1 + (1-\eta)\boldsymbol{\mu}_2$.*

That is, under the above setting with $\mathcal{R} = W_1$, the estimated mean of the inlier distribution, μ_1, coincides with the mean of the prior distribution, independently of the outlier distribution. However, when $\mathcal{R} = KL$, the estimated mean of the inlier distribution is sensitive to outliers.

3.2 Guarantees for (12) with Low-rank $\boldsymbol{\Sigma}_1$

We study the minimization problem (12) when $\boldsymbol{\Sigma}_1$ has a low rank and $\boldsymbol{\Sigma}_2 \in \mathcal{S}_{++}^K$. We fully analyze the cases where $\mathcal{R} = W_2$ and $\mathcal{R} = KL$; however, the case where $\mathcal{R} = W_1$ is difficult to analyze and compute. We first formulate results for both cases ($\mathcal{R} = W_2$ and $\mathcal{R} = KL$), and then clarify them. When $\mathcal{R} = W_2$, we assume that the prior distribution has zero mean vector $\boldsymbol{\mu}_0 = \mathbf{0}_K \in \mathbb{R}^K$ and covariance $\boldsymbol{\Sigma}_0 = \boldsymbol{I}_{K\times K} \in \mathbb{R}^{K\times K}$. We further denote by $\mathbf{1}_K$ the vector $(1, \cdots, 1) \in \mathbb{R}^K$. Similarly, we denote for any $n \in \mathbb{N}$, $\mathbf{0}_n$, $\mathbf{1}_n$, $\boldsymbol{I}_{n\times n}$. For vectors $\boldsymbol{a} \in \mathbb{R}^n$ and $\boldsymbol{b} \in \mathbb{R}^m$, we denote the concatenated vector in $\mathbb{R}^{n+m}$ by $(\boldsymbol{a}; \boldsymbol{b})$.

Proposition 3.2 *If κ, $K \in \mathbb{N}$, $K > \kappa \geq 1$, $\epsilon > 0$, $1 > \eta > \eta^\star := \frac{K-\kappa+\epsilon^2}{K-\kappa+2\epsilon^2}$, $u^\star := \left(\frac{(K-\kappa)(1-\eta)}{\epsilon^2(2\eta-1)}\right)^{\frac{1}{3}}$, where one can note that $\eta^\star > \frac{1}{2}$ and $u^\star \in (0,1)$, then the minimizer of (12) with $\mathcal{R} = W_2$ and the constraints that $\boldsymbol{\Sigma}_1$ is of rank κ and $\boldsymbol{\Sigma}_2$ is of rank K, satisfies $\mathbf{0}_K = u^\star\boldsymbol{\mu_2} + (1-u^\star)\boldsymbol{\mu_1}$, $\boldsymbol{\Sigma}_1 = \mathrm{diag}(\mathbf{1}_\kappa; \mathbf{0}_{K-\kappa})$ and $\boldsymbol{\Sigma}_2 = \mathrm{diag}(\mathbf{1}_\kappa; (u^\star)^{-2}\mathbf{1}_{K-\kappa})$. Moreover, $\|\boldsymbol{\mu}_1\|_2 = u^\star\epsilon$ and $\|\boldsymbol{\mu}_2\|_2 = (1-u^\star)\epsilon$.*

Proposition 3.3 *If κ, $K \in \mathbb{N}$, $K > \kappa \geq 1$, $\epsilon > 0$, $\eta > 0$, $\boldsymbol{\mu}_0$, $\boldsymbol{\mu}_1 \in \mathbb{R}^K$, $\boldsymbol{\Sigma}_0 \in \mathcal{S}_{++}^K$ and $\boldsymbol{\Sigma}_1 \in \mathcal{S}_{+}^K$, $\mathrm{rank}(\boldsymbol{\Sigma}_1) = \kappa$, then*

$$KL(\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)||\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)) = \infty.$$

Thus, the solution of (12) with $\mathcal{R} = KL$ and the additional constraints $\mathrm{rank}(\boldsymbol{\Sigma}_1) = \kappa$ and $\boldsymbol{\Sigma}_0 = \boldsymbol{I}$ is ill-posed.

Note that Proposition 3.2 implies that as $\eta \to 1$, $u^\star \to 0$. Hence for the inlier component $\boldsymbol{\mu}_1 \to \mathbf{0}_K$ as $\eta \to 1$ and $\boldsymbol{\Sigma}_1 = \mathrm{diag}(\mathbf{1}_\kappa; \mathbf{0}_{K-\kappa})$. Therefore, in the limit, the inlier distribution has the same mean as the prior distribution. Furthermore, its covariance is obtained by an appropriate projection of the covariance $\boldsymbol{\Sigma}_0$ onto a κ-dimensional subspace, independently of η. We similarly note that as $\eta \to 1$, $\boldsymbol{\Sigma}_2 \to \mathrm{diag}(\mathbf{1}_\kappa; \boldsymbol{\infty}_{K-k})$, so that the outliers disperse. The supplementary materials include the proof of Proposition 3.2 and a discussion that clarifies why the formulation and proof of Proposition 3.2 are not sufficient for inferring the effect of the W_1 minimization on MAW.

Proposition 3.3 implies that the KL divergence is unsuitable for low-rank covariance modeling as it leads to an infinite value in the optimization problem.

4 EXPERIMENTS

We describe the competing methods and experimental choices in §4.1. We report on the comparison with the competing methods in §4.2. We demonstrate the importance of the novel features of MAW in §4.3.

4.1 Competing Methods and Experimental Choices

We compared MAW with the following methods (descriptions and code links are in the supplementary materials): Deep Autoencoding Gaussian Mixture Model (DAGMM) (Zong et al., 2018), Deep Structured Energy-Based Models (DSEBMs) (Zhai et al., 2016), Isolation Forest (IF) (Liu et al., 2008), Local Outlier Factor (LOF) (Breunig et al., 2000), One-class Novelty Detection Using GANs (OCGAN) (Perera et al., 2019), One-Class SVM (OCSVM) (Heller et al., 2003) and RSR Autoencoder (RSRAE) (Lai et al., 2020).

We remark that IF, LOF and RSRAE were originally proposed for outlier detection and we thus apply their trained model for detecting novelties in the test data.

For MAW and the above four reconstruction-based methods, that is, DAGMM, DSEBMs, OCGAN and RSRAE, we use the following structure of encoders and decoders, which vary with the type of data (images or non-images). For non-images, which are mapped to feature vectors of dimension D, the encoder is a fully connected network with output channels $(32, 64, 128, 128 \times 4)$. The decoder is a fully connected network with output channels $(128, 64, 32, D)$, followed by a normalization layer at the end. For image datasets, the encoder has three convolutional layers with output channels $(32, 64, 128)$, kernel sizes $(5 \times 5, 5 \times 5, 3 \times 3)$ and strides $(2, 2, 2)$. Its output is flattened to lie in $\mathbb{R}^{128}$ and then mapped into a 128×4

dimensional vector using a dense layer (with output channels 128×4). The decoder of image datasets first applies a dense layer from $\mathbb{R}^2$ to $\mathbb{R}^{128}$ and then three deconvolutional layers with output channels $(64, 32, 3)$, kernel sizes $(3\times3, 5\times5, 5\times5)$ and strides $(2, 2, 2)$. For all experiments, the MAW discriminator is a fully connected network with size $(32, 64, 128, 1)$.

For MAW we set the following parameters, where additional details are in the supplementary materials. Intrinsic dimension: $d = 2$; mixture parameter: $\eta = 5/6$, sampling number: $T = 5$, and size of $\boldsymbol{A}$ (used for dimension reduction): 128×2. We further test the sensitivity of MAW to changes of the hyperparameters d and η in the supplementary materials. The code is available at `https://github.com/JCL823/MAW`.

4.2 Comparison of MAW with State-of-the-art Methods

We use six datasets for novelty detection: COVID-19 Radiography database (Chowdhury et al., 2020), CIFAR-10 (Krizhevsky, 2009), Caltech101 (Fei-Fei et al., 2004), Fashion MNIST (Xiao et al., 2017), KDDCUP-99 (Dua & Graff, 2017) and Reuters-21578 (Lewis, 1997). We distinguish between image datasets (COVID-19, CIFAR-10, Catlech101 and Fashion MNIST) and non-image datasets (KDDCUP-99 and Reuters-21578). We describe each dataset, common preprocessing procedures and choices of their largest clusters in the supplementary materials. Each dataset contains several clusters (3 for COVID-19, 10 for CIFAR-10, 11 largest ones for Caltech101, 10 for Fashion MNIST, 2 for KDDCUP-99 and 5 largest ones for Reuters-21578, respectively). We arbitrarily fix a class and uniformly sample N training inliers and N_{test} testing inliers from that class. We let $N = 160, 450, 100$, 300, 6000, 350 and $N_{\text{test}} = 60, 150, 100$, 60, 1200, 140 for COVID-19, CIFAR-10, Caltech101, Fashion MNIST, KDDCUP-99 and Reuters-21578, respectively. We fix c in $\{0.1, 0.2, 0.3, 0.4, 0.5\}$, and uniformly sample outliers for training from the rest of the clusters, while maintaining a fraction of c outliers per inliers. We also fix c_{test} in $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ and uniformly sample outliers from the rest of the clusters for testing, while maintaining a fraction of c_{test} per inliers.

Using all possible thresholds for the finite datasets, we compute the AUC (area under curve) and AP (average precision) scores, while considering the outliers as "positive". For each fixed $c = 0.1, 0.2, 0.3, 0.4, 0.5$ we average these results over the values of c_{test}, the different choices of an inlier cluster (among all possible clusters), and three runs with different random initializations for each of these choices. We also compute the corresponding standard deviations. We report these results in Fig. 2 and further specify numerical values in the supplementary materials. We observe state-of-the-art performance of MAW in all of these datasets. There are very special instances, where other methods perform better, for example, in Reuters-21578, DSEBMs performs slightly better than MAW and OCSVM has comparable performance. However, overall MAW is the most competitive method considering all instances. In the supplementary materials we compare the runtime of

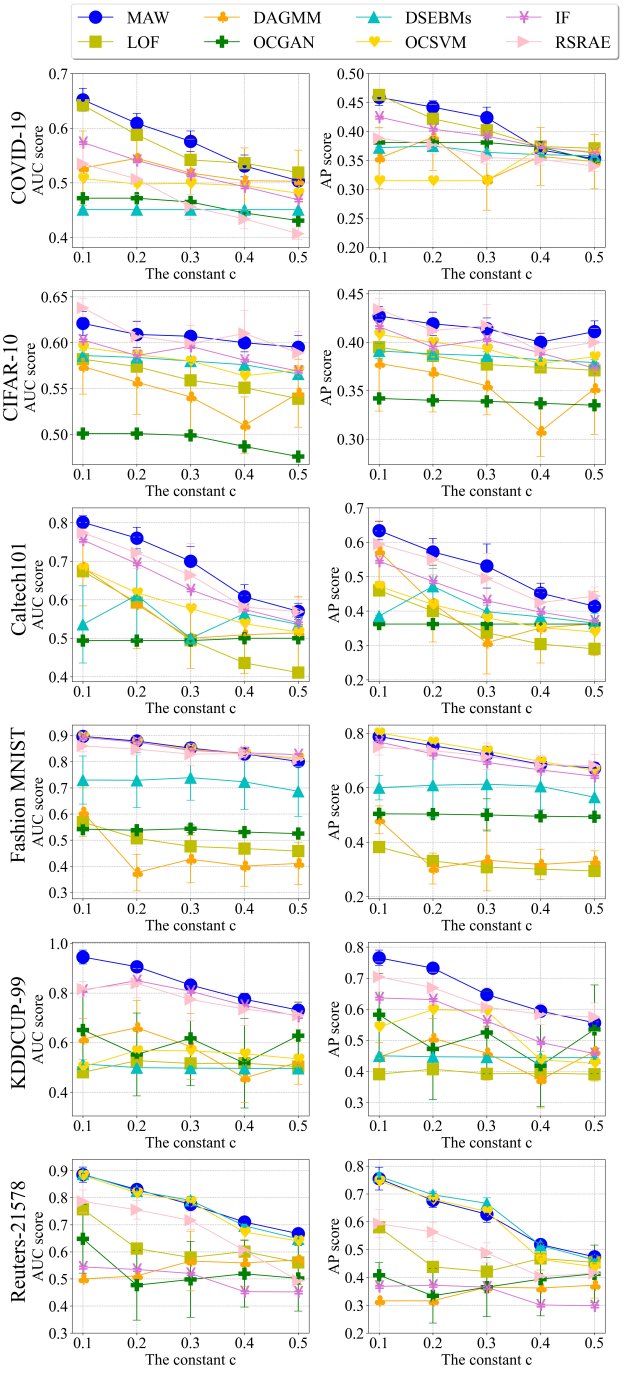

Figure 2: AUC (on left) and AP (on right) scores with training ratio of outliers per inliers $c = 0.1, 0.2, 0.3, 0.4$ and 0.5 for the six datasets.

MAW with benchmark methods and further study the accuracy of MAW in a different scenario, where the outliers of the training and test sets have different characteristics. We show that in this scenario MAW performs even better than the regular scenario.

4.3 Testing the Effect of the Novel Features of MAW

We experimentally validate the effect of the following features of MAW: the least absolute deviation for reconstruction, the W_1 metric for the regularization of the latent distribution, the GMM assumption, full covariance matrices resulting from the dimension reduction component, the lower rank constraint for the inlier mode and the use of a single mode prior distribution. To this end, we consider the following alternative models.

MAW-MSE: It replaces the least absolute deviation loss L_{VAE} with the common mean squared error (MSE).

MAW-KL divergence: It replaces the Wasserstein distance in (9) with the KL-divergence.

MAW-same rank: It uses the same rank d for both $\mathbf{\Sigma}_1^{(i)}$ and $\mathbf{\Sigma}_2^{(i)}$, instead of forcing $\mathbf{\Sigma}_1^{(i)}$ to have lower rank $d/2$.

MAW-single Gaussian: It replaces the GMM for the latent distribution with a single Gaussian with a full covariance matrix.

MAW-diagonal cov.: It replaces the full covariance matrices resulting from the dimension reduction component by diagonal covariances. Its encoder directly produces 2-dimensional means and diagonal covariances (one of rank 1 for the inlier mode and one of rank 2 for the outlier mode).

GMM prior: It replaces the single standard normal distribution prior with a bi-modal Gaussian distribution. One mode is a standard normal distribution in $\mathbb{R}^d$ and the other is Gaussian with zero mean and diagonal covariance matrix whose first $d/2$ diagonal elements are ones and the rest are zeros.

VAE: It has the same encoder and decoder structures as MAW. Instead of a dimension reduction component, it uses a dense layer which maps the output of the encoder to a 4-dimensional vector composed of a 2-dimensional mean and 2-dimensional diagonal covariance. This is common for a traditional VAE.

We compared the above 7 methods with MAW using two datasets: KDDCUP-99 and COVID-19 with training ratio of outliers per inliers $c = 0.1, 0.2, 0.3, 0.4$ and 0.5. We followed the experimental setting described in §4.1. Fig. 3 reports the averages and standard deviations of the computed AUC and AP scores, where the corresponding numerical values are further recorded in the supplementary materials. The results indicate a clear decrease of accuracy when missing any of the novel components of MAW

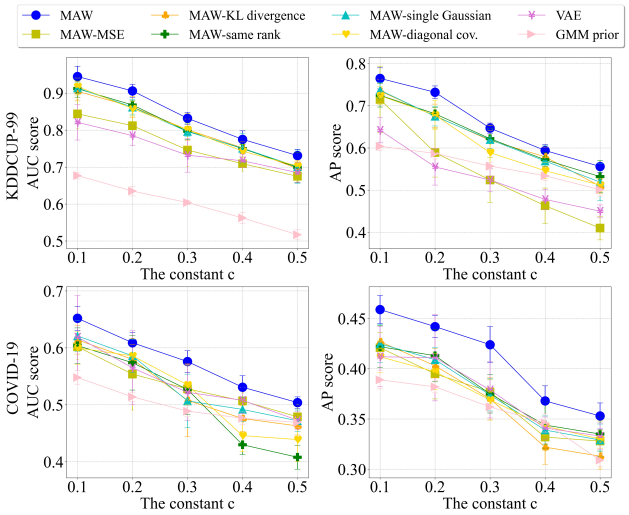

Figure 3: AUC (on left) and AP (on right) scores for variants of MAW (missing a novel component) with training ratio of outliers per inliers $c = 0.1, 0.2, 0.3, 0.4$ and 0.5, using KDDCUP-99 and COVID-19.

or using a standard VAE (i.e., "VAE"). Nevertheless, the use of a single diagonal matrix in "VAE" can help decrease the capacity of the latent distribution and thus "VAE" may perform better than the variants of MAW (but not MAW). In some cases, the variants of MAW show a rather poor performance and we believe it is due to the following reasons: modeling the prior as a Gaussian mixture in "GMM prior" does not help with outlier detection; the use of the full covariance in "MAW-single Gaussian" may result in high capacity; "MAW-MSE", "MAW-KL divergence" and "MAW-same rank" do not ensure either robustness or low-rank modeling for the inliers, and thus may significantly increase the capacity of the model so that learning from outliers is easier (especially for large c); and "MAW-diagonal cov." may limit the covariance of the outliers (though it is often at least comparable to VAE).

4.4 Further Validation of GMM

To further support our claim that the GMM is helpful for separating inliers and outliers in the latent space, we investigate the reconstruction errors of both MAW and MAW-single Gaussian of §4.3 (which replaces the GMM with a single Gaussian distribution with a full rank). We use the KDDCUP-99 dataset with 1,000 inliers and 300 outliers in the training set, where the initial training of MAW (or MAW-single Gaussian) is the same as in §4. In Fig. 4, we demonstrate the reconstruction error distribution of data points according to the following five scenarios.

1. **MAW, inliers and inlier distribution:** Apply the trained MAW (with the corrupted model) to the inliers of the training set, while using only the inlier mode in

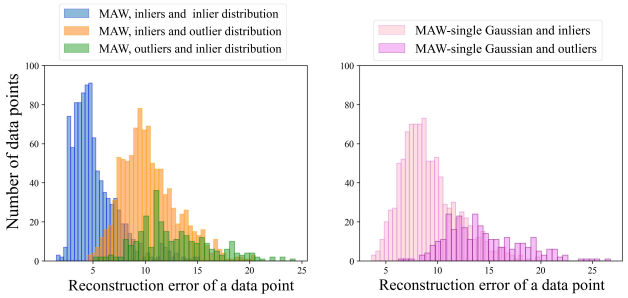

Figure 4: Demonstration of the distributions of the three types of reconstruction errors obtained with MAW (left) and the two types obtained with MAW-single Gaussian (right).

the latent code and compute the reconstruction error between the output and the input (the ℓ_2 norm of their difference).

2. **MAW, inliers and outlier distribution:** Same as case 1, but replace the inlier mode with the outlier mode.
3. **MAW, outliers and inlier distribution:** Same as case 1, but replace the inliers (input of MAW) with the outliers.
4. **MAW-single Gaussian and inliers:** Same as case 1, but replace MAW with MAW-single Gaussian.
5. **MAW-single Gaussian and outliers:** Same as 1, but replace the inliers (as input of the trained MAW-single Gaussian) with the outliers.

We can see from cases 1 and 2 above (which appear on the left of Fig. 4) that if we try to reconstruct the inliers, then the reconstruction errors with the outlier mode are higher than those with the inlier mode. In particular, it is obvious that the inlier and outlier modes are different and do not collapse. Although we did not supervisedly train the inlier and outlier modes, it seems that the inliers align well with the inlier distribution. Moreover, comparing cases 1 and 3 above (still left of Fig. 4), we can nicely distinguish between the distributions of the reconstruction errors of the inliers and the outliers. On the other hand, cases 4 and 5 (on the right of Fig. 4) indicate that when using MAW-single Gaussian instead of MAW, the distributions of reconstruction errors of the inliers and outliers are indistinguishable. This experiment thus demonstrates the effectiveness of the GMM of MAW in separating the inliers and outliers for this particular experiment.

5 CONCLUSION AND FUTURE WORK

We introduced MAW, a robust VAE-type framework for novelty detection that can tolerate high corruption of the training data. We proved that the Wasserstein distance used in MAW has better robustness to outliers and is more suitable to a low-dimensional inlier component than the KL divergence. We demonstrated state-of-the-art performance of MAW with a variety of datasets and experimentally validated that omitting any of the new ideas results in a significant decrease of accuracy.

We would like to indicate three limitations of MAW. First, there are some special instances, where other methods performed better than MAW, though overall MAW outperformed the rest of the methods. Second, MAW is slow. We expect that better implementation of its dimension reduction component can speed it up, so that it is as fast as other methods that use multiple neural networks. At last, MAW assumes the existence of both inlier and outlier modes for training (see assumptions of Props. 3.1 and 3.2). Indeed, one may check that the performance of MAW (and RSRAE) are not as competitive when $c = 0$. Since we assumed that the underlying distribution represented both inliers and outliers, we did not report such results.

MAW has practical applications of societal impact, such as medical diagnosis. One potential negative impact can arise if MAW identifies outliers due to their belonging to underrepresented groups. In the future, we would thus like to explore the overall fairness of MAW, possible fairer versions of it and the tradeoff between robustness and fairness in our theoretical setting.

Another future plan is to extend and test some of our ideas for the problem of robust generation, in particular, for building generative networks which are robust against adversarial training data. We also hope to further extend our theoretical guarantees. For example, two problems that currently seem intractable are the study of the W_1 version of Proposition 3.2 and of the minimizer in (15) (which is a weaker version of (12)).

Acknowledgements

This work was partially supported by NSF award DMS 2124913 and the Kunshan Municipal Government research funding.

References

Agueh, M. and Carlier, G. Barycenters in the Wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2): 904–924, 2011.

Akrami, H., Joshi, A. A., Li, J., and Leahy, R. M. Robust variational autoencoder. *arXiv preprint 1905.09961*, 2019.

An, J. and Cho, S. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1), 2015.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *Proceedings of the*

34th International Conference on Machine Learning, pp. 214–223. PMLR, 2017.

Aytekin, C., Ni, X., Cricri, F., and Aksu, E. Clustering and unsupervised anomaly detection with l 2 normalized deep auto-encoder representations. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6. IEEE, 2018.

Bengio, Y. and Monperrus, M. Non-local manifold tangent learning. In *Advances in Neural Information Processing Systems*, pp. 129–136, 2005.

Bergmann, P., Fauser, M., Sattlegger, D., and Steger, C. MVTec AD–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9592–9600, 2019.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104, 2000.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122, 2013.

Candès, E. J., Li, X., Ma, Y., and Wright, J. Robust principal component analysis? *J. ACM*, 58(3), 2011. ISSN 0004-5411.

Chalapathy, R. and Chawla, S. Deep learning for anomaly detection: A survey. *arXiv preprint 1901.03407*, 2019.

Chalapathy, R., Menon, A. K., and Chawla, S. Robust, deep and inductive anomaly detection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 36–51. Springer, 2017.

Chandola, V., Banerjee, A., and Kumar, V. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3): 1–58, 2009.

Chen, Y., Georgiou, T. T., and Tannenbaum, A. Optimal transport for Gaussian mixture models. *IEEE Access*, 7: 6269–6278, 2018.

Chowdhury, M. E. H., Rahman, T., Khandakar, A., Mazhar, R., Kadir, M. A., Mahbub, Z. B., Islam, K. R., Khan, M. S., Iqbal, A., Emadi, N. A., Reaz, M. B. I., and Islam, M. T. Can AI help in screening viral and COVID-19 pneumonia? *IEEE Access*, 8:132665–132676, 2020. doi: 10.1109/ACCESS.2020.3010287.

Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Dai, B., Wang, Y., Aston, J., Hua, G., and Wipf, D. Connections with robust pca and the role of emergent sparsity in variational autoencoder models. *The Journal of Machine Learning Research*, 19(1):1573–1614, 2018.

Daniel, T., Kurutach, T., and Tamar, A. Deep variational semi-supervised novelty detection. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

De La Torre, F. and Black, M. J. A framework for robust subspace learning. *International Journal of Computer Vision*, 54(1-3):117–142, 2003.

Ding, C., Zhou, D., He, X., and Zha, H. R1-PCA: rotational invariant l_1-norm principal component analysis for robust subspace factorization. In *Proceedings of the 23rd international conference on Machine learning*, pp. 281–288. ACM, 2006.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Eduardo, S., Nazábal, A., Williams, C. K. I., and Sutton, C. Robust variational autoencoders for outlier detection and repair of mixed-type data. In *AISTATS*, 2020.

Fan, H., Zhang, F., Wang, R., Xi, L., and Li, Z. Correlation-aware deep generative model for unsupervised anomaly detection. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 688–700, 2020.

Fei-Fei, L., Fergus, R., and Perona, P. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. In *2004 Conference on Computer Vision and Pattern Recognition Workshop*, pp. 178–178, 2004.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

Golan, I. and El-Yaniv, R. Deep anomaly detection using geometric transformations. In *Advances in Neural Information Processing Systems*, pp. 9758–9769, 2018.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.

Heller, K., Svore, K., Keromytis, A. D., and Stolfo, S. One class support vector machines for detecting anomalous windows registry accesses, 2003.

Hershey, J. R. and Olsen, P. A. Approximating the Kullback Leibler divergence between gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pp. IV–317. IEEE, 2007.

Huang, C. and Tagare, H. D. Robust W-estimators for cryo-EM class means. *IEEE Transactions on Image Processing*, 25(2):893–906, 2015.

Ilonen, J., Paalanen, P., Kamarainen, J.-K., and Kalviainen, H. Gaussian mixture pdf in one-class classification: computing and utilizing confidence values. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 2, pp. 577–580. IEEE, 2006.

Jiang, H., Wang, H., Hu, W., Kakde, D., and Chaudhuri, A. Fast incremental SVDD learning algorithm with the Gaussian kernel. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:3991–3998, 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*, 2015.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Kolouri, S., Rohde, G. K., and Hoffmann, H. Sliced wasserstein distance for learning gaussian mixture models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3427–3436, 2018.

Kolouri, S., Nadjahi, K., Simsekli, U., Badeau, R., and Rohde, G. Generalized sliced wasserstein distances. *Advances in neural information processing systems*, 32, 2019.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Lai, C.-H., Zou, D., and Lerman, G. Robust subspace recovery layer for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2020.

Lee, K., Lee, K., Lee, H., and Shin, J. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pp. 7167–7177, 2018.

Lerman, G. and Maunu, T. Fast, robust and non-convex subspace recovery. *Information and Inference: A Journal of the IMA*, 7(2):277–336, 2017.

Lerman, G. and Maunu, T. An overview of robust subspace recovery. *Proceedings of the IEEE*, 106(8):1380–1410, 2018.

Lerman, G. and Zhang, T. l_p-recovery of the most significant subspace among multiple subspaces with outliers. *Constructive Approximation*, 40:329–385, 2014.

Lerman, G., McCoy, M. B., Tropp, J. A., and Zhang, T. Robust computation of linear models by convex relaxation. *Foundations of Computational Mathematics*, 15 (2):363–410, 2015.

Lewis, D. Reuters-21578 text categorization test collection. *Distribution 1.0, AT&T Labs-Research*, 1997.

Liu, F. T., Ting, K. M., and Zhou, Z.-H. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422. IEEE, 2008.

Lopuhaa, H. P. and Rousseeuw, P. J. Breakdown points of affine equivariant estimators of multivariate location and covariance matrices. *Ann. Statist.*, 19(1):229–248, 03 1991. doi: 10.1214/aos/1176347978.

Makhzani, A., Shlens, J., Jaitly, N., and Goodfellow, I. Adversarial autoencoders. In *International Conference on Learning Representations Workshop*, 2016.

Maunu, T. and Lerman, G. Robust subspace recovery with adversarial outliers. *arXiv preprint 1904.03275*, 2019.

Maunu, T., Zhang, T., and Lerman, G. A well-tempered landscape for non-convex robust subspace recovery. *Journal of Machine Learning Research*, 20(37):1–59, 2019.

McCoy, M. and Tropp, J. A. Two proposals for robust PCA using semidefinite programming. *Electronic Journal of Statistics*, 5:1123–1160, 2011.

Mescheder, L., Nowozin, S., and Geiger, A. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *International conference on machine learning*, pp. 2391–2400. PMLR, 2017.

Miolane, N., Poitevin, F., Li, Y.-T., and Holmes, S. Estimation of orientation and camera parameters from cryo-electron microscopy images with variational autoencoders and generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 970–971, 2020.

Moya, M. M. and Hush, D. R. Network constraints and multi-objective optimization for one-class classification. *Neural Networks*, 9(3):463–474, 1996.

Northcutt, C. G., Athalye, A., and Mueller, J. Pervasive label errors in test sets destabilize machine learning benchmarks. In *Proceedings of the 35th Conference on Neural Information Processing Systems Track on Datasets and Benchmarks*, December 2021.

Panaretos, V. M. and Zemel, Y. Statistical aspects of wasserstein distances. *Annual review of statistics and its application*, 6:405–431, 2019.

Perera, P., Nallapati, R., and Xiang, B. OCGAN: One-class novelty detection using gans with constrained latent representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2898–2906, 2019.

Perera, P., Oza, P., and Patel, V. M. One-class classification: A survey. *arXiv preprint 2101.03064*, 2021.

Peyré, G. and Cuturi, M. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Pidhorskyi, S., Almohsen, R., and Doretto, G. Generative probabilistic novelty detection with adversarial autoencoders. In *Advances in neural information processing systems*, pp. 6822–6833, 2018.

Pimentel, M. A., Clifton, D. A., Clifton, L., and Tarassenko, L. A review of novelty detection. *Signal Processing*, 99:215–249, 2014.

Pol, A. A., Berger, V., Germain, C., Cerminara, G., and Pierini, M. Anomaly detection with conditional variational autoencoders. In *2019 18th IEEE international conference on machine learning and applications (ICMLA)*, pp. 1651–1657. IEEE, 2019.

Rajaraman, A. and Ullman, J. D. *Mining of massive datasets*. Cambridge University Press, 2011.

Sabokrou, M., Khalooei, M., Fathy, M., and Adeli, E. Adversarially learned one-class classifier for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3379–3388, 2018.

Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., and Platt, J. C. Support vector method for novelty detection. In *Advances in neural information processing systems*, pp. 582–588, 2000.

Shyu, M.-L., Chen, S.-C., Sarinnapakorn, K., and Chang, L. A novel anomaly detection scheme based on principal component classifier. In *ICDM Foundation and New Direction of Data Mining workshop*, 2003.

Tolstikhin, I., Bousquet, O., Gelly, S., and Schoelkopf, B. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018.

Vaswani, N. and Narayanamurthy, P. Static and dynamic robust PCA and matrix completion: A review. *Proceedings of the IEEE*, 106(8):1359–1379, 2018.

Wang, K. and Lan, H. Robust support vector data description for novelty detection with contaminated data. *Engineering Applications of Artificial Intelligence*, 91: 103554, 2020.

Watson, G. A. *Some Problems in Orthogonal Distance and Non-Orthogonal Distance Regression*. Defense Technical Information Center, 2001.

Wright, J., Ganesh, A., Rao, S., Peng, Y., and Ma, Y. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems*, pp. 2080–2088, 2009.

Xian, Y., Sharma, S., Schiele, B., and Akata, Z. f-vaegan-d2: A feature generating framework for any-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10275–10284, 2019.

Xiao, A. T., Tong, Y. X., and Zhang, S. False-negative of RT-PCR and prolonged nucleic acid conversion in COVID-19: Rather than recurrence. *Journal of Medical Virology*, 2020.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint 1708.07747*, 2017.

Xiao, Y., Wang, H., Xu, W., and Zhou, J. Robust one-class SVM for fault detection. *Chemometrics and Intelligent Laboratory Systems*, 151:15 – 25, 2016.

Xu, H., Caramanis, C., and Sanghavi, S. Robust PCA via outlier pursuit. *IEEE Trans. Information Theory*, 58(5): 3047–3064, 2012. doi: 10.1109/TIT.2011.2173156.

Xu, H., Chen, W., Zhao, N., Li, Z., Bu, J., Li, Z., Liu, Y., Zhao, Y., Pei, D., Feng, Y., Chen, J., Wang, Z., and Qiao, H. Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications. In *Proceedings of the 2018 World Wide Web Conference*, pp. 187–196, 2018.

Ye, F. and Bors, A. G. Infovaegan: learning joint interpretable representations by information maximization and maximum likelihood. In *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 749–753. IEEE, 2021.

Zhai, S., Cheng, Y., Lu, W., and Zhang, Z. Deep structured energy based models for anomaly detection. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pp. 1100–1109. PMLR, 2016.

Zhang, C., Li, S., Zhang, H., and Chen, Y. VELC: A new variational autoencoder based model for time series anomaly detection. *arXiv preprint 1907.01702*, 2019.

Zhang, T. and Lerman, G. A novel M-estimator for robust PCA. *Journal of Machine Learning Research*, 15 (1):749–808, 2014.

Zhang, T., Szlam, A., and Lerman, G. Median K-flats for hybrid linear modeling with many outliers. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pp. 234–241. IEEE, 2009.

Zhou, C. and Paffenroth, R. C. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 665–674, 2017.

Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., and Chen, H. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018.

SUPPLEMENTARY MATERIALS

We include additional explanations, proofs, demonstrations and experiments as follows: §A further clarifies MAW and its implementation; §B examines the sensitivity of MAW to hyperparameters; §C compares the runtime of MAW with benchmark methods; §D extends the previous numerical studies to a different type of outliers; §E extends our theoretical discussion and proves all the stated propositions; §F reviews the details of the benchmark methods; §G reviews the details of the datasets; and §H provides numerical tables for the results plotted in the different figures.

A ADDITIONAL EXPLANATIONS AND IMPLEMENTATION DETAILS OF MAW

In §A.1 we review the ELBO function and explain our robust version of ELBO. The basic mechanism of MAW is clarified in §A.2. Additional implementation details of MAW are in §A.3. At last, §A.4 provides algorithmic boxes for training MAW and applying it for novelty detection.

A.1 Obtaining $\mathcal{L}_{\text{MAW}}$ by Modifying ELBO

A standard VAE framework would minimize the expected KL-divergence from $p(\mathbf{z}|\mathbf{x})$ to $q(\mathbf{z}|\mathbf{x})$ in $\mathcal{Q}$, where the expectation is taken over $p(\mathbf{x})$. By Bayes' rule this is equivalent to maximizing the evidence lower bound (ELBO):

$$\text{ELBO}(q) = \mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z}) - \mathbb{E}_{p(\mathbf{x})} KL(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))\ .$$

The first term of ELBO is the reconstruction likelihood. Its second term restricts the deviation of $q(\mathbf{z}|\mathbf{x})$ from $p(\mathbf{z})$ and can be viewed as a regularization term. $\mathcal{L}_{\text{MAW}}$ is a more robust version of ELBO with a different regularization. Recall that for $p \geq 1$, we denote by W_p the p-Wasserstein distance in $\mathbb{R}^D$. For two probability distributions, μ, ν on $\mathbb{R}^D$,

$$W_p(\mu,\nu) = \left(\inf_{\pi\in\Pi(\mu,\nu)} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\pi} \|\mathbf{x}-\mathbf{y}\|_2^p\right)^{1/p},$$

where $\Pi(\mu,\nu)$ is the set of joint distributions with μ and ν as marginals. MAW replaces $\mathbb{E}_{p(\mathbf{x})} KL(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))$ with $W_1(q(\mathbf{z}), p(\mathbf{z}))$. We remark that the W_1 distance cannot be computed between $q(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{z})$ and $\mathcal{L}_{\text{MAW}}$ thus practically replaces $q(\mathbf{z}|\mathbf{x})$ with its expected distribution, $q(\mathbf{z}) = \mathbb{E}_{p(\mathbf{x})} q(\mathbf{z}|\mathbf{x})$ (or a discrete approximation of this).

We emphasize that $\mathcal{L}_{\text{MAW}}$ is not necessarily a lower bound of the likelihood. The W_1 distance in $\mathcal{L}_{\text{MAW}}$ can rather be understood as a regularization involving the estimated posterior and prior distribution.

A.2 Insights on the Mechanism of MAW

We explain the basic mechanism of MAW for unsupervised alignment of the inliers with the inlier mode of the latent distribution. Since we do not have labels for the training set, we cannot supervisedly determine the inlier mode. Nevertheless, the robust losses (the least absolute deviation and the W_1 distance) guide the estimation of the inlier mode as they help in ignoring the effect of the outliers. Least absolute deviation metrics have been shown to be robust to outliers in special mathematical settings (Lopuhaa & Rousseeuw, 1991; Lerman & Maunu, 2018; Lai et al., 2020). The robustness of the Wasserstein distance within a mathematical setting was studied in §3 of the main text. Here we would like to provide some intuition on how the complex procedure of MAW succeeds by using these robust metrics.

Assume that the inliers are sampled from a distribution on a low-dimensional manifold that can be encoded by a Gaussian on a low-dimensional latent space. Assume further that the outliers are arbitrary, but their percentage is smaller. Given these assumptions, MAW aims to model the mixture component of the inliers in the latent space as a Gaussian with low-rank covariance (and that of the outliers as a Gaussian with full-rank covariance).

In order to provide some technical intuition for this model and show that it can fit the assumed data, let us suppose on the contrary that during training, inliers and outliers are assigned to the wrong modes, and show that this can either not happen or will be corrected.

We first assume a case of collapse during training, where both the inliers and outliers are modeled (in the latent space) by a Gaussian distribution with a low-rank covariance. In this case, the W_1 distance is minimized over a smaller set (due to the constraint on the rank of the outlier mode) and thus the loss is increased.

We next assume another case of collapse during training, where both the inliers and outliers are modeled (in the latent space) by a full-rank Gaussian. In this case it is most likely that the minimizer for the inliers will be full-rank, and thus due to the assumed low-dimensional structure of the inliers, it will result in an increase of the reconstruction error.

At last, assume that during training the inliers are modeled (in the latent space) by a Gaussian with full-rank covariance and the outliers are modeled (in the latent space) by a Gaussian with a low-rank covariance. One can note that this will increase the reconstruction loss.

A.3 Additional Implementation Details of MAW

All NNs were implemented with TensorFlow (available at `tensorflow.org`) and trained for 100 epochs with batch size 128. We apply batch normalization to each layer of any NN. For the VAE-structure of MAW, we use Adam with a learning rate of 0.0005. For the WGAN-type discriminator of MAW, we perform RMSprop (Bengio & Monperrus, 2005) with a learning rate of 0.0005, following the recommendation of Arjovsky et al. (2017) for WGAN. For all experiments, the MAW discriminator is a fully connected network of size $(32, 64, 128, 1)$. The matrix $\boldsymbol{A}$ and the network parameters for encoders, decoders and discriminators are initialized by the Glorot uniform initializer (Glorot & Bengio, 2010).

The implementation details of the reconstruction-based methods are similar to those of MAW. In particular, we optimized using Adam (Kingma & Ba, 2015) with a learning rate of 0.0005.

A.4 Algorithms for MAW

Algorithms 1 and 2 describe the training and application of MAW for novelty detection. We denote by $\boldsymbol{\theta}$, $\boldsymbol{\varphi}$ and $\boldsymbol{\delta}$ the trainable parameters of the encoder $\mathcal{E}$, decoder $\mathcal{D}$ and discriminator $\mathcal{D}is$, respectively. Recall that $\boldsymbol{A}$ includes the trained parameters of the dimension reduction component.

Algorithm 1 Training MAW

Input: Training data $\{\mathbf{x}^{(i)}\}_{i=1}^{L}$; initialized parameters $\boldsymbol{\theta}$, $\boldsymbol{\varphi}$ and $\boldsymbol{\delta}$ of $\mathcal{E}$, $\mathcal{D}$ and $\mathcal{D}is$, respectively; initialized $\boldsymbol{A}$; weight η; number of epochs; batch size I; sampling number T; learning rate α

Output: Trained parameters $\boldsymbol{\theta}$, $\boldsymbol{\varphi}$ and $\boldsymbol{A}$

for each epoch **do**
  for each batch $\{\mathbf{x}^{(i)}\}_{i\in\mathcal{I}}$ **do**
    $\boldsymbol{\mu}_{0,1}^{(i)}, \boldsymbol{\mu}_{0,2}^{(i)}, \boldsymbol{s}_{0,1}^{(i)}, \boldsymbol{s}_{0,2}^{(i)} \leftarrow \mathcal{E}(\mathbf{x}^{(i)})$
    $\boldsymbol{\mu}_j^{(i)} \leftarrow \boldsymbol{A}^{\mathrm{T}}\boldsymbol{\mu}_{0,j}^{(i)},\ \boldsymbol{M}_j^{(i)} \leftarrow \boldsymbol{A}^{\mathrm{T}}\mathrm{diag}(\boldsymbol{s}_{0,j}^{(i)})\boldsymbol{A},\ j=1,2$
    Compute $\tilde{\boldsymbol{M}}_1^{(i)}$ according to (6) and (7)
    $\boldsymbol{\Sigma}_1^{(i)} \leftarrow \tilde{\boldsymbol{M}}_1^{(i)}\tilde{\boldsymbol{M}}_1^{(i)\mathrm{T}},\ \boldsymbol{\Sigma}_2^{(i)} \leftarrow \boldsymbol{M}_2^{(i)}\boldsymbol{M}_2^{(i)\mathrm{T}}$
    for $t = 1, \cdots, T$ **do**
      sample a batch $\{\mathbf{z}_{\text{gen}}^{(i,t)}\}_{i\in\mathcal{I}} \sim \eta\mathcal{N}(\boldsymbol{\mu}_1^{(i)}, \boldsymbol{\Sigma}_1^{(i)}) + (1-\eta)\mathcal{N}(\boldsymbol{\mu}_2^{(i)}, \boldsymbol{\Sigma}_2^{(i)})$
      sample a batch $\{\mathbf{z}_{\text{hyp}}^{(i,t)}\}_{i\in\mathcal{I}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
    end for
    $(\boldsymbol{\theta}, \boldsymbol{A}, \boldsymbol{\varphi}) \leftarrow (\boldsymbol{\theta}, \boldsymbol{A}, \boldsymbol{\varphi}) - \alpha\nabla_{(\boldsymbol{\theta},\boldsymbol{A},\boldsymbol{\varphi})}L_{\mathrm{VAE}}(\boldsymbol{\theta}, \boldsymbol{A}, \boldsymbol{\varphi})$ according to (8)
    $\boldsymbol{\delta} \leftarrow \boldsymbol{\delta} - \alpha\nabla_{\boldsymbol{\delta}}L_{W_1}(\boldsymbol{\delta})$ according to (10)
    $\boldsymbol{\delta} \leftarrow \mathrm{clip}(\boldsymbol{\delta}, [-1, 1])$
    $(\boldsymbol{\theta}, \boldsymbol{A}) \leftarrow (\boldsymbol{\theta}, \boldsymbol{A}) - \alpha\nabla_{(\boldsymbol{\theta},\boldsymbol{A})}L_{\mathrm{GEN}}(\boldsymbol{\theta}, \boldsymbol{A})$ according to (11)
  end for
end for

Algorithm 2 Applying MAW to novelty detection

Input: Test data $\{\mathbf{y}^{(j)}\}_{j=1}^{N}$; sampling number T; trained MAW model; threshold ϵ_{T}; similarity $S(\cdot,\cdot)$
Output: Binary labels for novelty for each $j = 1, \dots, N$

for $j = 1, \dots, N$ **do**
  $\boldsymbol{\mu}_{0,1}^{(j)}, \boldsymbol{s}_{0,1}^{(j)} \leftarrow \mathcal{E}(\mathbf{y}^{(j)})$
  $\boldsymbol{\mu}_{1}^{(j)} \leftarrow \boldsymbol{A}^{\mathrm{T}}\boldsymbol{\mu}_{0,1}^{(j)},\ \boldsymbol{M}_{1}^{(j)} \leftarrow \boldsymbol{A}^{\mathrm{T}}\mathrm{diag}(\boldsymbol{s}_{0,1}^{(j)})\boldsymbol{A}$
  Compute $\tilde{\boldsymbol{M}}_{1}^{(j)}$ according to (6) and (7)
  $\boldsymbol{\Sigma}_{1}^{(j)} \leftarrow \tilde{\boldsymbol{M}}_{1}^{(j)}\tilde{\boldsymbol{M}}_{1}^{(j)\mathrm{T}}$
  for $t = 1, \cdots, T$ **do**
    sample $\mathbf{z}_{\text{in}}^{(j,t)} \sim \mathcal{N}(\boldsymbol{\mu}_{1}^{(j)}, \boldsymbol{\Sigma}_{1}^{(j)})$
    $\tilde{\mathbf{y}}^{(j,t)} \leftarrow \mathcal{D}\left(\mathbf{z}_{\text{in}}^{(j,t)}\right)$
    compute $S(\mathbf{y}^{(j)}, \tilde{\mathbf{y}}^{(j,t)})$
  end for
  $S^{(j)} \leftarrow T^{-1}\sum_{t=1}^{T} S(\mathbf{y}^{(j)}, \tilde{\mathbf{y}}^{(j,t)})$
  if $S^{(j)} \geq \epsilon_{\mathrm{T}}$ **then**
    $\mathbf{y}^{(j)}$ is a normal example
  else
    $\mathbf{y}^{(j)}$ is a novelty
  end if
end for

B SENSITIVITY TO SOME HYPERPARAMETERS

We examine sensitivity to choices of the intrinsic dimension (see §B.1) and the mixture parameter (see §B.2).

B.1 Sensitivity to the Intrinsic Dimension

Our default value of the intrinsic dimension is $d = 2$. Here we study the sensitivity of our numerical results to the following choices intrinsic dimensions: $d = 2$, 4, 8, 16, 32 and 64, while using the KDDCUP-99 and COVID-19 datasets. The training ratio of outliers per inliers c are in $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. We compute the AUC and AP scores averaged over the testing ratios of outliers per inliers, $c_{\text{test}} = 0.1$, 0.3, 0.5, 0.7 and 0.9, and over three runs of the same setting. Fig. 5 reports the averaged results and their standard deviations, which are indicated by error bars.

We can see that larger intrinsic dimensions generally result in better performances. However, the improvement is not significant and not consistent for smaller dimensions. Furthermore, higher dimensions require more substantial computational efforts for training.

B.2 Sensitivity to the Mixture Parameter

The default value of the mixture parameter η is $5/6$. Here we study the sensitivity of the accuracy of MAW to the mixture parameters: $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 5/6, 0.9\}$. We use $5/6 \approx 0.83$, instead of the nearby value 0.8, since it was already tested for MAW. The training ratios of outliers per inliers are 0.1, 0.2, 0.3, 0.4 and 0.5. Following the same procedure of §B, we average the AUC and AP scores for both KDDCUP-99 and COVID-19. We report them in Fig. 6.

We notice that the AUC and AP scores mildly increase as η increases (though they may slightly decrease at 0.9). It seems that MAW learns well the inlier mode with a sufficiently large inlier weight, where the variation in the accuracy as a function of η is not large in general.

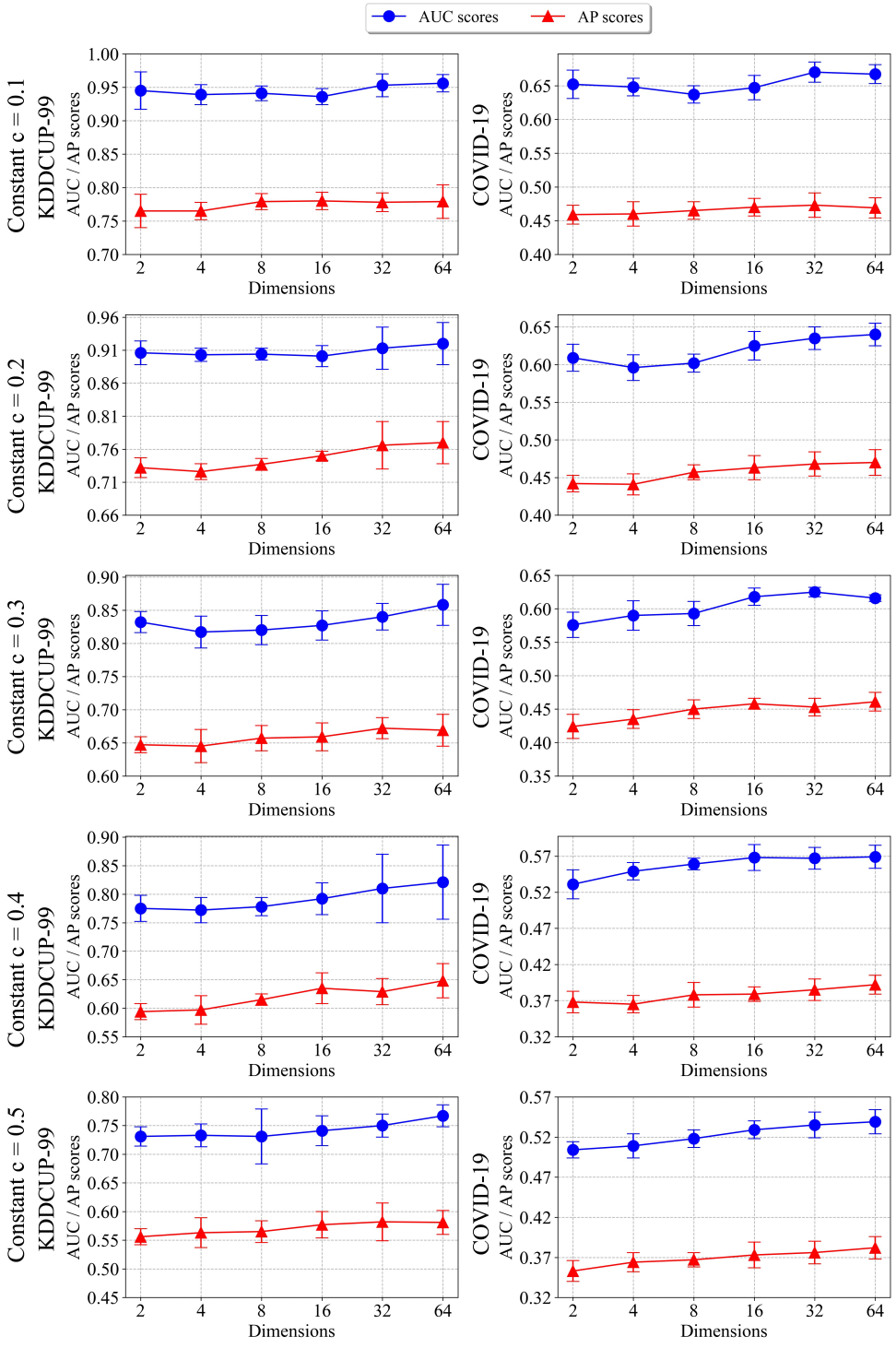

Figure 5: AUC and AP scores with intrinsic dimensions $d = 2$, 4, 8, 16, 32 and 64 for KDDCUP-99 (on the left) and COVID-19 (on the right), where $c \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$

.

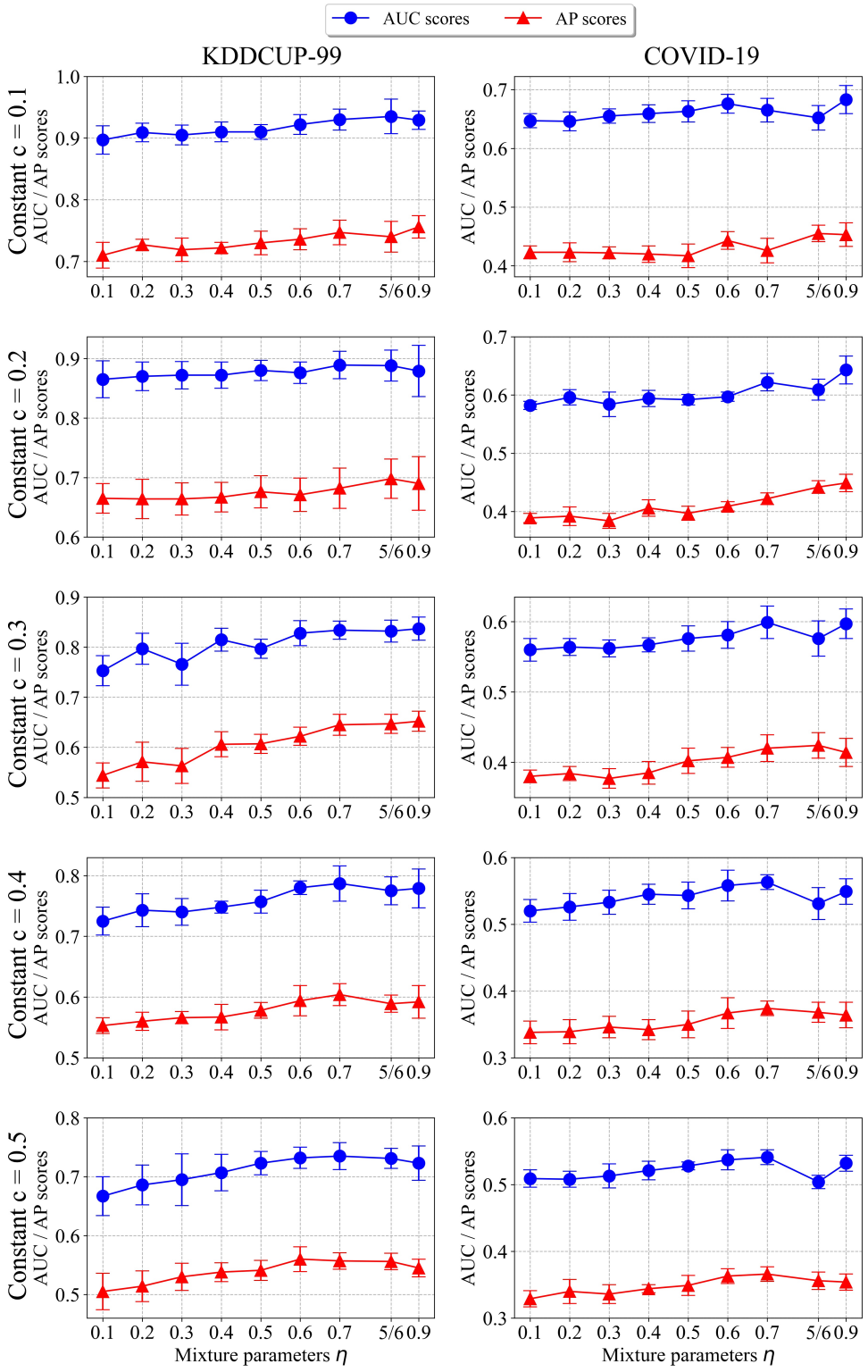

Figure 6: AUC and AP scores with mixture parameters $\eta = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 5/6$ and 0.9 for KDDCUP-99 (on the left) and COVID-19 (on the right). From the top to the bottom row, the training ratios of outliers per inliers are $c = 0.1, 0.2, 0.3, 0.4$ and 0.5, respectively.

C RUNTIME COMPARISON

Table 1 summarizes runtimes of all the above experiments with $c = 0.3$. The initially computed runtimes are times measured for completing single experiments with a single epoch. The table averages each such runtime over the different classes and different outlier ratios for testing. We note that LOF, OCSVM and IF are faster than the rest of the methods since they do not require training neural networks. Among the neural-networks-based methods, RSRAE is the fastest and OCGAN, DAGMM and MAW are the slowest. Indeed, RSRAE has a single autoencoder and OCGAN, DAGMM and MAW contain several neural networks. Another possible reason for the relative slowness of MAW is due to its dimension reduction component, whose implementation in TensorFlow seems to be computationally expensive. However, it seems to help achieve competitive performance in detecting outliers. We plan to investigate a more efficient implementation of the dimension reduction component in the future.

Table 1: Runtimes (in seconds) of competing methods when the training ratio of outliers per inliers is $c = 0.3$.

Methods	COVID-19	CIFAR-10	Caltech101	Fashion MNIST	KDDCUP-99	Reuters-21578
LOF	0.30 ± 0.17	3.98 ± 0.13	0.24 ± 0.01	16.31 ± 1.01	3.23 ± 0.04	17.91 ± 1.98
OCSVM	0.17 ± 0.07	2.22 ± 0.09	0.12 ± 0.00	8.34 ± 2.36	9.08 ± 0.05	8.74 ± 1.47
IF	0.43 ± 0.01	1.86 ± 0.12	0.39 ± 0.01	2.86 ± 0.37	1.67 ± 0.03	10.54 ± 1.89
RSRAE	4.31 ± 0.45	8.49 ± 0.77	5.69 ± 0.36	23.69 ± 0.39	40.18 ± 0.33	6.22 ± 0.25
DSEBMs	48.30 ± 3.45	66.57 ± 2.35	147.15 ± 0.32	151.02 ± 7.67	216.02 ± 4.34	74.28 ± 2.09
OCGAN	182.79 ± 2.53	313.28 ± 0.13	679.44 ± 4.62	250.51 ± 0.24	2035.83 ± 8.34	343.02 ± 7.42
DAGMM	99.44 ± 5.76	134.36 ± 9.12	504.11 ± 11.31	353.65 ± 14.57	396.44 ± 7.65	177.30 ± 3.56
MAW	136.42 ± 0.16	1871.22 ± 16.03	217.32 ± 0.25	1441.97 ± 15.123	3166.62 ± 12.12	255.85 ± 2.97

D EXPERIMENTS WITH DIFFERENT TYPES OF OUTLIERS

In this section, we test the performance of MAW and the benchmark methods when the training and test sets are corrupted by outliers with different structures. We generate a dataset, which we call "Mix Caltech101", in the following way. We fix the largest class of Caltech101 (containing airplane images) as the inlier class and randomly split it into the training inlier class (68.75%) and testing inlier class (31.25%). We form the training set by corrupting the training inlier class with random samples from the ten classes of CIFAR-10 (Krizhevsky, 2009) with training ratio of outliers per inliers $c \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. For the test set, we corrupt the testing inlier class by "tile images" from MVTech dataset (Bergmann et al., 2019) with testing ratio of outliers per inliers c_{test} in $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. The rest of the settings of the experiments are identical to the description in §4.2 of the main text. We present the AUC and AP scores and their standard deviations in Fig. 7.

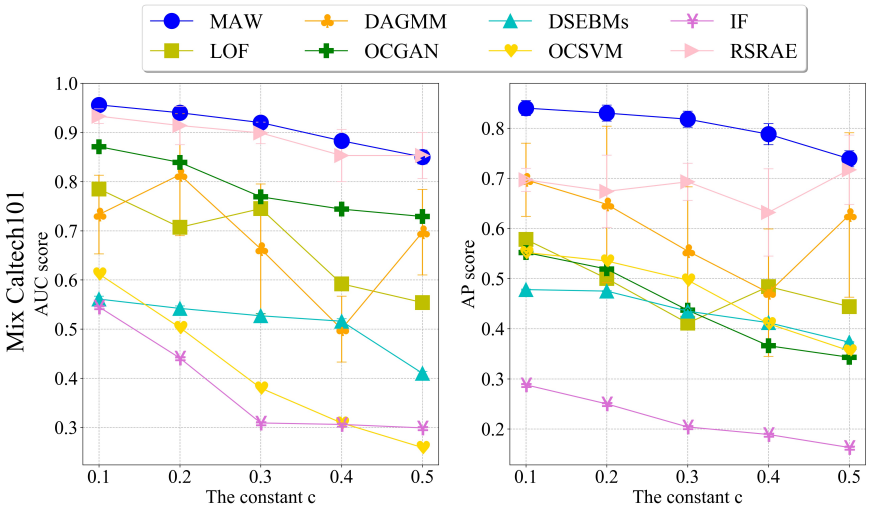

Figure 7: AUC and AP scores with training ratio of outliers per inliers $c \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ for the Mix Caltech101 dataset.

Clearly, the competitive advantage of MAW is also noticeable in this setting. We note that OCSVM, the traditional distance-based method, and IF, the traditional density-based method, perform poorly in this scenario, whereas they performed well in our original setting.

E ADDITIONAL THEORETICAL GUARANTEES FOR THE W_1 MINIMIZATION

In §E.1 we fully motivate our focus on studying (12) in order to understand the advantage of the use of the Wasserstein distance over the KL divergence in the framework of MAW. In §E.2 we prove Proposition 3.1. In §E.3, we discuss a possible deviation of the clean theory of Proposition 3.2 from practice. In §E.4 we prove Proposition 3.2 and in §E.5 we prove Proposition 3.3.

E.1 Motivation for Studying (12)

The implementation of any VAE or its variants, such as AAE, WAE and MAW, requires the optimization of a regularization penalty $\mathcal{R}$, which measures the discrepancy between the latent and prior distributions. This penalty is typically the KL divergence, though one may use appropriate metrics such as W_2 or W_1. Thus one needs to minimize

$$\mathcal{R}\left(\frac{1}{L}\sum_{i=1}^{L} q(\mathbf{z}|\mathbf{x}^{(i)}), p(\mathbf{z})\right) \tag{13}$$

over the variational family $\mathcal{Q} = \{q(\mathbf{z}|\mathbf{x})\}$ indexed by some parameters. Here L is the batch size of the input data and $\sum_{i=1}^{L} q(\mathbf{z}|\mathbf{x}^{(i)})$ is its observed aggregated distribution.

Since the explicit expressions of the regularization measurements between aggregated distributions are unknown, it is not feasible to study the minimizer of (13). We thus consider the following approximation of (13):

$$\sum_{i=1}^{L} \frac{1}{L}\mathcal{R}\left(q(\mathbf{z}|\mathbf{x}^{(i)}), p(\mathbf{z})\right). \tag{14}$$

We can minimize one term of this sum at a time, that, is minimize $\mathcal{R}\left(q(\mathbf{z}|\mathbf{x}), p(\mathbf{z})\right)$ over $\mathcal{Q}$. This minimization strategy is common in the study of the Wasserstein barycenter problem (Agueh & Carlier, 2011; Peyré & Cuturi, 2019; Chen et al., 2018).

One of the underlying assumptions of MAW is that the prior distribution $p(\mathbf{z})$ is Gaussian and $q(\mathbf{z}|\mathbf{x})$ is a Gaussian mixture. That is, $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu_0}, \boldsymbol{\Sigma}_0)$ and $q(\mathbf{z}|\mathbf{x}) = \eta\mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + (1-\eta)\mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$. This gives rise to the following minimization problem

$$\min_{\boldsymbol{\mu}_1,\boldsymbol{\mu}_2\in\mathbb{R}^K;\boldsymbol{\Sigma}_1,\boldsymbol{\Sigma}_2\in\mathcal{S}_+^K} \mathcal{R}\Big(\eta\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + (1-\eta)\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2), \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)\Big). \tag{15}$$

Similarly to approximating (13) by (14), we approximate (15) by (12). We remark that in (12) we further assume that there is a sufficiently small threshold $\epsilon > 0$ for which $\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 \geq \epsilon$. This is a reasonable assumption since, in practice, if $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are very close, the reconstruction loss will be large.

E.2 Proof of Proposition 3.1

Recall that $\boldsymbol{\mu}_0 \in \mathbb{R}^K$ is the mean of the prior Gaussian, $\epsilon > 0$ is the fixed separation parameter for the means of the two modes and $\eta > 1/2$ is the fixed mixture parameter. For $i = 0, 1, 2$, we denote the Gaussian probability distribution by $\nu_i = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$. Since in our setting $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2$, we denote the common covariance matrix in $\mathcal{S}_{++}^K$ by $\boldsymbol{\Sigma}$. That is, $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_i$ for $i = 0, 1, 2$.

We first analyze the solution of (12) with $\mathcal{R} = W_p$, where $p \geq 1$, and then analyze the solution of (12) with $\mathcal{R} = KL$.

The case $\mathcal{R} = W_p, p \geq 1$: We follow the next three steps to prove that the minimizer of (12) satisfies $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_0$.

Step I: We prove that

$$\begin{aligned} W_p(\nu_i, \nu_0) &\equiv W_p(\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}), \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma})) \\ &= \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_0\|_2 \quad \text{for } p \geq 1 \text{ and } i = 1, 2\,. \end{aligned} \tag{16}$$

First, we note that using the definition of $W_p, p \geq 1$ and the common notation $\Pi(\nu_i, \nu_0)$ for the distribution on $\mathbb{R}^K \times \mathbb{R}^K$ with marginals ν_i and ν_0

$$\begin{aligned} W_p^p(\nu_i, \nu_0) &= \inf_{\pi \in \Pi(\nu_i, \nu_0)} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\pi} \|\mathbf{x} - \mathbf{y}\|_2^p \\ &\geq \inf_{\pi \in \Pi(\nu_i, \nu_0)} \left\|\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\pi}\mathbf{x} - \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\pi}\mathbf{y}\right\|_2^p \\ &= \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_0\|_2^p \ , \end{aligned} \tag{17}$$

where the inequality follows from the fact that $\|.\|_2^p$ is convex and from Jensen's inequality.

On the other hand, for $i = 1$ or $i = 2$, let $\mathbf{x}^*$ be an arbitrary random vector with distribution ν_i, and let $\mathbf{y}^* = \mathbf{x}^* - \boldsymbol{\mu}_i + \boldsymbol{\mu}_0$. The distribution of $\mathbf{y}^*$ is Gaussian with mean $\boldsymbol{\mu}_0$ and covariance $\boldsymbol{\Sigma}_i$, that is, this distribution is ν_0. Let π^* be the joint distribution of the random variables $\mathbf{x}^*$ and $\mathbf{y}^*$. We note that π^* is in $\Pi(\nu_i, \nu_0)$ and that

$$\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\pi^*} \|\mathbf{x} - \mathbf{y}\|_2^p = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\pi^*} \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_0\|_2^p = \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_0\|_2^p \ .$$

Therefore,

$$\begin{aligned} W_p^p(\nu_i, \nu_0) &= \inf_{\pi \in \Pi(\nu_i, \nu_0)} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\pi} \|\mathbf{x} - \mathbf{y}\|_2^p \\ &\leq \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\pi^*} \|\mathbf{x} - \mathbf{y}\|_2^p = \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_0\|_2^p \ . \end{aligned} \tag{18}$$

The combination of (17) and (18) immediately yields (16).

Step II: We prove that (12) with $\mathcal{R} = W_p$, $p \geq 1$, is equivalent to

$$\min_{\substack{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathbb{R}^K; \\ \text{s.t. } \boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \text{:colinear} \\ \& \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 \geq \epsilon}} \eta \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\|_2 + (1-\eta) \|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_0\|_2 \,. \tag{19}$$

We first note that (12) with $\mathcal{R} = W_p$, $p \geq 1$ is equivalent to

$$\min_{\substack{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathbb{R}^K \\ \text{s.t. } \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 \geq \epsilon}} \eta \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\|_2 + (1-\eta) \|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_0\|_2 \,. \tag{20}$$

Indeed, this is a direct consequence of the expression derived in step I for $\mathcal{R}$ in this case. It is thus left to show that if $\boldsymbol{\mu}_1'$, $\boldsymbol{\mu}_2' \in \mathbb{R}^K$ minimize (20), then we can construct $\widetilde{\boldsymbol{\mu}_1'}$, $\widetilde{\boldsymbol{\mu}_2'} \in \mathbb{R}^K$ that are colinear with $\boldsymbol{\mu}_0$ and also minimize (20).

For any $\boldsymbol{\mu_1}$ and $\boldsymbol{\mu_2}$ in $\mathbb{R}^K$ with $\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 \geq \epsilon$ and for the given $\boldsymbol{\mu}_0 \in \mathbb{R}^K$, we define $\tilde{\boldsymbol{\mu}}_0$, $\tilde{\boldsymbol{\mu}}_1$ and $\tilde{\boldsymbol{\mu}}_2 \in \mathbb{R}^K$ and demonstrate them in Fig. 8. The point $\tilde{\boldsymbol{\mu}}_0$ is the projection of $\boldsymbol{\mu}_0$ onto $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$ and $\tilde{\boldsymbol{\mu}}_i := \boldsymbol{\mu}_i + \boldsymbol{\mu}_0 - \tilde{\boldsymbol{\mu}}_0$ for $i = 1, 2$. We observe the following properties, which can be proved by direct calculation, though Fig. 8 also clarifies them:

$$\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_0\|_2 \geq \|\tilde{\boldsymbol{\mu}}_i - \boldsymbol{\mu}_0\|_2 \text{ for } i = 1, 2,$$

and consequently,

$$\eta \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\|_2 + (1-\eta) \|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_0\|_2 \geq \eta \|\tilde{\boldsymbol{\mu}}_1 - \boldsymbol{\mu}_0\|_2 + (1-\eta) \|\tilde{\boldsymbol{\mu}}_2 - \boldsymbol{\mu}_0\|_2 \,; \tag{21}$$

$$\|\tilde{\boldsymbol{\mu}}_1 - \tilde{\boldsymbol{\mu}}_2\|_2 = \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 \geq \epsilon; \tag{22}$$

and

$$\tilde{\boldsymbol{\mu}}_1, \ \tilde{\boldsymbol{\mu}}_2, \ \text{ and } \boldsymbol{\mu}_0 \ \text{ are colinear.} \tag{23}$$

Clearly, the combination of (21), (22) and (23) concludes the proof of step II. That is, it implies that if $\boldsymbol{\mu}_1'$, $\boldsymbol{\mu}_2' \in \mathbb{R}^K$ minimize (20), then $\widetilde{\boldsymbol{\mu}_1'}$ and $\widetilde{\boldsymbol{\mu}_2'}$ defined above are colinear with $\boldsymbol{\mu}_0$ and also minimize (20).

Step III: We directly solve (19) and consequently (12) with $\mathcal{R} = W_p$, $p \geq 1$. Due to the colinearity constraint in (12), we can write

$$\boldsymbol{\mu_0} = (1+t)\boldsymbol{\mu_1} - t\boldsymbol{\mu_2} \ \text{ for } t \in \mathbb{R}. \tag{24}$$

The objective function in (19) can then be written as

$$\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 \left(\eta|t| + (1-\eta)|1+t|\right) \geq \epsilon \left(\eta|t| + (1-\eta)|1+t|\right),$$

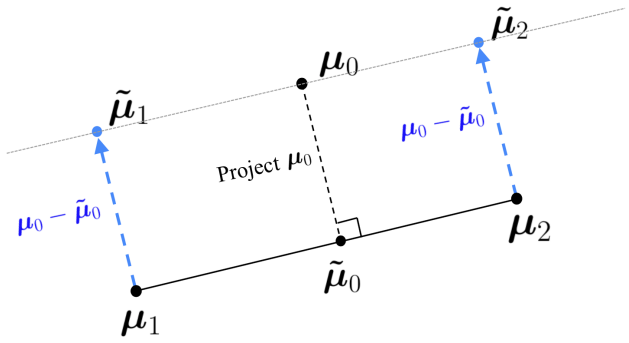

Figure 8: Illustration of the points $\tilde{\boldsymbol{\mu}}_0$, $\tilde{\boldsymbol{\mu}}_1$ and $\tilde{\boldsymbol{\mu}}_2$ and their properties.

where equality is achieved if and only if $\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 = \epsilon$. We thus define $r(t) = \eta|t| + (1-\eta)|1+t|$ and note that

$$r(t) = \begin{cases} t + (1-\eta), & t \geq 0 \\ (1-2\eta)t + (1-\eta), & 0 \geq t \geq -1 \\ -t + (\eta - 1), & -1 \geq t \end{cases}$$

and its derivative is

$$r'(t) = \begin{cases} 1, & t > 0 \\ 1 - 2\eta, & 0 > t > -1 \\ -1, & -1 > t. \end{cases}$$

The above expressions for r and r' and the assumption that $\eta > 1/2$ imply that $r(t)$ is increasing when $t > 0$, decreasing when $t < 0$ and $r(0) = 1 - \eta < \eta = r(1)$. Thus r has a global minimum at $t = 0$. Hence, it follows from (24) that the minimizer of (12), and equivalently (12) with $\mathcal{R} = W_p$, $p \geq 1$ satisfies $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_0$.

The case $\mathcal{R} = KL$: We prove that the solution of (12) with $\mathcal{R} = KL$ satisfies $\boldsymbol{\mu}_0 = \eta\boldsymbol{\mu}_1 + (1-\eta)\boldsymbol{\mu}_2$. We practically follow similar steps as the proof above.

Step I: We derive an expression for $KL(\nu_i||\nu_0)$, where $i = 1, 2$. We use the following general formula, which holds for the case where $\boldsymbol{\Sigma}_0$, $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ are general covariance matrices in $\mathcal{S}_{++}^K$ (see e.g., (2) in (Hershey & Olsen, 2007)):

$$KL(\nu_i||\nu_0) = \frac{1}{2}\left(\log\frac{\det\boldsymbol{\Sigma}_0}{\det\boldsymbol{\Sigma}_i} - K + \mathrm{tr}(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}_i) + (\boldsymbol{\mu}_i - \boldsymbol{\mu}_0)^{\mathrm{T}}\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_0)\right). \tag{25}$$

Since in our setting $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$, this expression has the simpler form:

$$KL(\nu_i||\nu_0) = \frac{1}{2}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_0)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_0).$$

Step II: We reformulate the optimization problem. The above step implies that (12) with $\mathcal{R} = KL$ can be written as

$$\min_{\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 \geq \epsilon} \quad \eta(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) + (1-\eta)(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_0)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_0),$$

or equivalently,

$$\min_{\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 \geq \epsilon} \quad \eta\left\|\boldsymbol{\Sigma}^{-\frac{1}{2}}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)\right\|_2^2 + (1-\eta)\left\|\boldsymbol{\Sigma}^{-\frac{1}{2}}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_0)\right\|_2^2. \tag{26}$$

We express the eigenvalue decomposition of $\boldsymbol{\Sigma}^{-1}$ as $\boldsymbol{\Sigma}^{-1} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{T}}$, where $\boldsymbol{\Lambda} \in \mathcal{S}_+^K$, and $\boldsymbol{U}$ is an orthogonal matrix. Applying the change of variables $\boldsymbol{\mu}_i^{'} = \boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{\mu}_i$ for $i = 0, 1, 2$, we rewrite (26) as

$$\min_{\left\|\boldsymbol{\mu}_1^{'} - \boldsymbol{\mu}_2^{'}\right\|_2 \geq \epsilon} \quad \eta\left\|\boldsymbol{\mu}_1^{'} - \boldsymbol{\mu}_0^{'}\right\|_2^2 + (1-\eta)\left\|\boldsymbol{\mu}_2^{'} - \boldsymbol{\mu}_0^{'}\right\|_2^2. \tag{27}$$

At last, applying the same colinearity argument as above (supported by Fig. 8) we conclude the following equivalent formulation of (27):

$$\min_{\substack{\boldsymbol{\mu}_0^{'},\boldsymbol{\mu}_1^{'},\boldsymbol{\mu}_2^{'} \text{ are colinear} \\ \& \left\|\boldsymbol{\mu}_1^{'}-\boldsymbol{\mu}_2^{'}\right\|_2 \geq \epsilon}} \quad \eta \left\|\boldsymbol{\mu}_1^{'} - \boldsymbol{\mu}_0^{'}\right\|_2^2 + (1-\eta) \left\|\boldsymbol{\mu}_2^{'} - \boldsymbol{\mu}_0^{'}\right\|_2^2 \tag{28}$$

Step III: We directly solve (28). Due to the colinearity constraint, we can write

$$\boldsymbol{\mu_0}^{'} = (1+t)\boldsymbol{\mu_1}^{'} - t\boldsymbol{\mu_2}^{'} \text{ for } t \in \mathbb{R} \tag{29}$$

and express the objective function of (28) as

$$\left\|\boldsymbol{\mu}_1^{'} - \boldsymbol{\mu}_2^{'}\right\|_2^2 \left(\eta t^2 + (1-\eta)(1+t)^2\right) \geq \epsilon^2 \left(\eta t^2 + (1-\eta)(1+t)^2\right),$$

where equality is achieved if and only if $\|\boldsymbol{\mu}_1' - \boldsymbol{\mu}_2'\|_2 = \epsilon$. We thus define $r(t) = \eta t^2 + (1-\eta)(1+t)^2$ and note that $r'(t) = 2(t+(1-\eta))$ and $r''(t) = 2$, and thus conclude that $r(t)$ obtains its global minimum at $t = \eta - 1$. This observation and (29) imply that the minimizers $\boldsymbol{\mu_1}$ and $\boldsymbol{\mu_2}$ of (12) with $\mathcal{R} = KL$ satisfy $\boldsymbol{\mu_0} = \eta\boldsymbol{\mu_1} + (1-\eta)\boldsymbol{\mu_2}$.

E.3 Some Remarks on Proposition 3.2

We clarify why the statement and proof of the proposition are not sufficient for explaining the effect of the W_1 optimization on MAW. We note that the inlier and outlier covariances, $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$, obtained by Proposition 3.2, are diagonal. Furthermore, the proof of Proposition 3.2 clarifies that the underlying minimization problem of this proposition may assume without loss of generality that the inlier and outlier covariances are diagonal (see e.g., (31)). On the other hand, the numerical results in §4.3 of the main text support the use of full covariances, instead of diagonal covariance. Nonetheless, we claim that the full covariances of MAW come naturally from the dimension reduction component of MAW. This component also contains trainable parameters for the covariances and they will affect the weights of the encoder, that is, will affect both the W_1 minimization and the reconstruction loss. Thus the analysis of the W_1 minimization component is not sufficient for inferring the whole behavior of MAW. For tractability purposes, the minimization in (12) ignores the dimension reduction component. For completeness we remark that there are two other differences between the use of (12) in Proposition 3.2 and the way it arises in MAW that may possibly also result in the advantage of using full covariance in MAW. First of all, the minimization in Proposition 3.2 uses $\mathcal{R} = W_2$, whereas MAW uses $\mathcal{R} = W_1$, which we find intractable when using the rest of the setting of Proposition 3.2. Second of all, (12) with $\mathcal{R} = W_1$ is an approximation of the minimization of $W_1\left(\frac{1}{L}\sum_{i=1}^{L} q(\mathbf{z}|\mathbf{x}^{(i)}), p(\mathbf{z})\right)$ (see §E.1 for explanation), which is also intractable (even if one uses $\mathcal{R} = W_2$).

E.4 Proof of Proposition 3.2

We follow the same steps of the proof of Proposition 3.1.

Step I: We immediately verify the formula

$$W_2(\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \mathcal{N}(\mathbf{0}, \boldsymbol{I})) = \sqrt{\|\boldsymbol{\mu}_i\|_2^2 + \left\|\boldsymbol{\Sigma}_i^{\frac{1}{2}} - \boldsymbol{I}\right\|_F^2} \text{ for } i = 1, 2. \tag{30}$$

We use the following general formula, which holds for the case where $\boldsymbol{\Sigma}_0$, $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ are general covariance matrices in $\mathcal{S}_+^K$ (see e.g., (4) in (Panaretos & Zemel, 2019)): For $i = 1, 2$

$$W_2^2(\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)) = \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_0\|_2^2 + \mathrm{tr}(\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_0 - 2(\boldsymbol{\Sigma}_i^{\frac{1}{2}}\boldsymbol{\Sigma}_0\boldsymbol{\Sigma}_i^{\frac{1}{2}})^{\frac{1}{2}}).$$

Indeed, (30) is obtained as a direct consequence of (E.4) using the identity

$$\mathrm{tr}\left(\boldsymbol{\Sigma}_i + \boldsymbol{I} - 2\boldsymbol{\Sigma}_i^{\frac{1}{2}}\right) = \mathrm{tr}\left(\left(\boldsymbol{\Sigma}_i^{\frac{1}{2}} - \boldsymbol{I}\right)^2\right) = \left\|\boldsymbol{\Sigma}_i^{\frac{1}{2}} - \boldsymbol{I}\right\|_F^2.$$

Step II: We reformulate the underlying minimization problem in two different stages. We first claim that the minimizer of (12) with $\mathcal{R} = W_2$ and the constraint that $\mathbf{\Sigma}_1$ is of rank κ and $\mathbf{\Sigma}_2$ is of rank K can be expressed as the minimizer of

$$\min_{\substack{\boldsymbol{\mu}_1,\boldsymbol{\mu}_2\in\mathbb{R}^K \text{s.t. } \|\boldsymbol{\mu}_1-\boldsymbol{\mu}_2\|_2=\epsilon,\\ \mathbf{\Sigma}_1,\mathbf{\Sigma}_2 \text{ diagonal in } \mathbb{R}^{K\times K}\\ \& \text{ rank}(\mathbf{\Sigma}_1)=\kappa,\ \text{rank}(\mathbf{\Sigma}_2)=K}} \left[\eta\sqrt{\|\boldsymbol{\mu}_1\|_2^2+\left\|\mathbf{\Sigma}_1^{\frac{1}{2}}-\boldsymbol{I}\right\|_F^2}+(1-\eta)\sqrt{\|\boldsymbol{\mu}_2\|_2^2+\left\|\mathbf{\Sigma}_2^{\frac{1}{2}}-\boldsymbol{I}\right\|_F^2}\right]. \tag{31}$$

In view of (12) and (30) we only need to prove that the minimizer of (31) is the same if one removes the constraint that $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ are both diagonal matrices and require instead that they are in $\in \mathcal{S}_+^K$. This is easy to show. Indeed, if for $i = 1$ or $i = 2$, $\mathbf{\Sigma}_i \in \mathcal{S}_+^K$, then it can be diagonalized as follows: $\mathbf{\Sigma}_i = \boldsymbol{U}_i^{\mathrm{T}}\mathbf{\Lambda}_i\boldsymbol{U}_i$, where $\mathbf{\Lambda}_i \in \mathcal{S}_+^K$ is diagonal and $\boldsymbol{U}_i$ is orthogonal. Hence, $\mathbf{\Sigma}_i^{\frac{1}{2}} = \boldsymbol{U}_i^{\mathrm{T}}\mathbf{\Lambda}_i^{\frac{1}{2}}\boldsymbol{U}_i$ and

$$\left\|\mathbf{\Sigma}_i^{\frac{1}{2}}-\boldsymbol{I}\right\|_F^2=\left\|\boldsymbol{U}_i^{\mathrm{T}}\mathbf{\Lambda}_i^{\frac{1}{2}}\boldsymbol{U}_i-\boldsymbol{I}\right\|_F^2=\left\|\boldsymbol{U}_i^{\mathrm{T}}(\mathbf{\Lambda}_i^{\frac{1}{2}}-\boldsymbol{I})\boldsymbol{U}_i\right\|_F^2=\left\|\mathbf{\Lambda}_i^{\frac{1}{2}}-\boldsymbol{I}\right\|_F^2.$$

Consequently,

$$W_2(\mathcal{N}(\boldsymbol{\mu}_i,\mathbf{\Sigma}_i),\mathcal{N}(\mathbf{0},\boldsymbol{I}))=W_2(\mathcal{N}(\boldsymbol{\mu}_i,\mathbf{\Lambda}_i),\mathcal{N}(\mathbf{0},\boldsymbol{I})) \text{ for } i=1,2\,,$$

and the above claim is concluded.

Next, we vectorize the minimization problem in (31) as follows. We denote by $\mathbb{R}_+$ the set of positive real numbers. Let $\boldsymbol{b}$ be a general vector in $\mathbb{R}_+^K$, $\boldsymbol{a}'$ be a general vector in $\mathbb{R}_+^\kappa$ and $\boldsymbol{a} := (\boldsymbol{a}';\mathbf{0}_{K-\kappa}) \in \mathbb{R}^K$. Given, the constraints on $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$, we can parametrize the diagonal elements of $\mathbf{\Sigma}_1^{\frac{1}{2}}$ and $\mathbf{\Sigma}_2^{\frac{1}{2}}$ by $\boldsymbol{a}$ and $\boldsymbol{b}$, that is, we set $\mathbf{\Sigma}_1^{\frac{1}{2}} = \text{diag}(\boldsymbol{a})$ and $\mathbf{\Sigma}_2^{\frac{1}{2}} = \text{diag}(\boldsymbol{b})$. The objective function of (31) can then be written as

$$\eta\sqrt{\|\boldsymbol{\mu_1}\|_2^2+\|\boldsymbol{a}-\mathbf{1}_K\|_2^2}+(1-\eta)\sqrt{\|\boldsymbol{\mu_2}\|_2^2+\|\boldsymbol{b}-\mathbf{1}_K\|_2^2}.$$

Combining this last expression and the same colinearity argument as in the proof of Proposition 3.1 in §E.2 (supported by Fig. 8), (31) is equivalent to

$$\min_{\substack{\boldsymbol{\mu}_1,\boldsymbol{\mu}_2\in\mathbb{R}^K,\\ \boldsymbol{b}\in\mathbb{R}_+^K,\ \boldsymbol{a}'\in\mathbb{R}_+^\kappa,\ \boldsymbol{a}=(\boldsymbol{a}';\mathbf{0}_{K-\kappa}),\\ (\boldsymbol{\mu}_1;\boldsymbol{a}),(\boldsymbol{\mu}_2;\boldsymbol{b}),(\mathbf{0}_K;\mathbf{1}_K) \text{ are colinear}\\ \&\|\boldsymbol{\mu}_1-\boldsymbol{\mu}_2\|_2=\epsilon}} \left[\eta\left\|(\boldsymbol{\mu}_1;\boldsymbol{a})-(\mathbf{0}_K;\mathbf{1}_K)\right\|_2+(1-\eta)\left\|(\boldsymbol{\mu}_2;\boldsymbol{b})-(\mathbf{0}_K;\mathbf{1}_K)\right\|_2\right]. \tag{32}$$

Step III: We solve (32). By the colinearity constraint, we can write $(\mathbf{0}_K;\mathbf{1}_K) = u(\boldsymbol{\mu}_2;\boldsymbol{b}) - (u-1)(\boldsymbol{\mu}_1;\boldsymbol{a})$, where $u \in \mathbb{R}$. We thus obtain that

$$\begin{aligned}(\boldsymbol{\mu}_2;\boldsymbol{b})-(\mathbf{0}_K;\mathbf{1}_K)&=(u-1)\left((\boldsymbol{\mu}_1;\boldsymbol{a})-(\boldsymbol{\mu}_2;\boldsymbol{b})\right)\\ (\boldsymbol{\mu}_1;\boldsymbol{a})-(\mathbf{0}_K;\mathbf{1}_K)&=u\left((\boldsymbol{\mu}_1;\boldsymbol{a})-(\boldsymbol{\mu}_2;\boldsymbol{b})\right).\end{aligned} \tag{33}$$

Furthermore, denoting the coordinates of $\boldsymbol{a}'$ and $\boldsymbol{b}$ by $\{a_i\}_{i=1}^\kappa$ and $\{b_i\}_{i=1}^K$, we similarly obtain that

$$\begin{aligned}\mathbf{0}_K&=u\boldsymbol{\mu}_2-(u-1)\boldsymbol{\mu}_1\\ 1&=ub_i-(u-1)a_i,\quad 1\le i\le\kappa\\ 1&=ub_i,\quad d+1\le i\le K\end{aligned} \tag{34}$$

The last two of equations imply that

$$\sum_{i=1}^{\kappa}(a_i-b_i)^2=\frac{\|\mathbf{1}_\kappa-\boldsymbol{a}'\|_2^2}{u^2}$$

and

$$\sum_{i=\kappa+1}^{K}b_i^2=\frac{K-\kappa}{u^2}.$$

Combining (30), (33) and the above two equations, we rewrite the objective function of (32) as follows:

$$
\begin{aligned}
&\big(\eta|u| + |u-1|(1-\eta)\big) \times \|(\boldsymbol{\mu}_1; \boldsymbol{a}) - (\boldsymbol{\mu}_2; \boldsymbol{b})\|_2 \\
=&\big(\eta|u| + |u-1|(1-\eta)\big) \times \sqrt{\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2^2 + \sum_{i=1}^{\kappa}(a_i - b_i)^2 + \sum_{i=\kappa+1}^{K} b_i^2} \\
\geq& \left(\eta|u| + |u-1|(1-\eta)\right) \times \sqrt{\epsilon^2 + \frac{\|\mathbf{1}_\kappa - \boldsymbol{a}'\|_2^2}{u^2} + \frac{K-\kappa}{u^2}} \\
=&\left\{(K-\kappa)\left((1-\eta)\left|\frac{u-1}{u}\right| + \eta\right)^2 + \epsilon^2\big(\eta|u| + |u-1|(1-\eta)\big)^2 + \|\mathbf{1}_\kappa - \boldsymbol{a}'\|_2^2\left((1-\eta)\left|\frac{u-1}{u}\right| + \eta\right)^2\right\}^{1/2}, \quad (35)
\end{aligned}
$$

where equality is achieved if and only if $\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 = \epsilon$. One can make the following two observations: $u = 0$ does not yield a minimizer of (32), and for any $u \neq 0$, (35) obtains its minimum at $\boldsymbol{a}' = \mathbf{1}_\kappa$. In view of these observations and the derivation above, we define

$$
f(u) := (K-\kappa)\left((1-\eta)\left|\frac{u-1}{u}\right| + \eta\right)^2 + \epsilon^2\left(\eta|u| + |u-1|(1-\eta)\right)^2, \quad (36)
$$

and note that (32) is equivalent to

$$
\min_{u \neq 0} \quad \sqrt{f(u)}. \quad (37)
$$

We rewrite $f(u)$ as

$$
f(u) = \begin{cases}
(K-\kappa)\left(\dfrac{u-1}{u}(1-\eta) + \eta\right)^2 + \epsilon^2\Big(\eta u + (1-\eta)(u-1)\Big)^2, & u \geq 1 \\
(K-\kappa)\left(\dfrac{1-u}{u}(1-\eta) + \eta\right)^2 + \epsilon^2\Big(\eta u + (1-\eta)(1-u)\Big)^2, & 1 \geq u > 0 \\
(K-\kappa)\left(\dfrac{u-1}{u}(1-\eta) + \eta\right)^2 + \epsilon^2\Big(\eta u + (1-\eta)(u-1)\Big)^2, & 0 > u
\end{cases}
$$

We denote

$$
r_1(u) := (K-\kappa)\left(\frac{u-1}{u}(1-\eta) + \eta\right)^2 + \epsilon^2\Big(\eta u + (1-\eta)(u-1)\Big)^2
$$

and

$$
r_2(u) := (K-\kappa)\left(\frac{1-u}{u}(1-\eta) + \eta\right)^2 + \epsilon^2\Big(\eta u + (1-\eta)(1-u)\Big)^2.
$$

Their derivatives are

$$
r_1'(u) = \frac{2}{u^3}\left(u - (1-\eta)\right)\left(\epsilon^2 u^3 + (K-\kappa)(1-\eta)\right)
$$

and

$$
r_2'(u) = \frac{2}{u^3}\Big((2\eta-1)u + (1-\eta)\Big) \times \left(\epsilon^2(2\eta-1)u^3 - (K-\kappa)(1-\eta)\right).
$$

These expressions for r_1' and r_2' imply that the critical points for r_1 are

$$
u_{r_1}^{(1)} = 1 - \eta \ \text{ and } \ u_{r_1}^{(2)} = -\left(\frac{(K-\kappa)(1-\eta)}{\epsilon^2}\right)^{\frac{1}{3}}
$$

and the critical points for r_2 are

$$
u_{r_2}^{(1)} = -\left(\frac{1-\eta}{2\eta-1}\right) \ \text{ and } \ u_{r_2}^{(2)} = \left(\frac{(K-\kappa)(1-\eta)}{\epsilon^2(2\eta-1)}\right)^{\frac{1}{3}}.
$$

We note that r_1 is increasing on $(u_{r_1}^{(2)}, 0) \cup (u_{r_1}^{(1)}, \infty)$ and decreasing on $(-\infty, u_{r_1}^{(2)}) \cup (0, u_{r_1}^{(1)})$. On the other hand, r_2 is increasing on $(u_{r_2}^{(1)}, 0) \cup (u_{r_2}^{(2)}, \infty)$ and decreasing on $(-\infty, u_{r_2}^{(1)}) \cup (0, u_{r_2}^{(2)})$. Since $\eta > \eta^\star = \dfrac{K-\kappa+\epsilon^2}{K-\kappa+2\epsilon^2}$, $u_{r_2}^{(2)} \in (0,1)$. The derivative of f with respect to u is

$$f'_u(u) = \begin{cases} r'_1(u), & u > 0 \\ r'_2(u), & 1 > u > 0 \\ r'_1(u), & 0 > u. \end{cases}$$

So $f(\cdot)$ is increasing on $(u_{r_1}^{(2)}, 0) \cup (u_{r_2}^{(2)}, \infty)$ and decreasing on $(-\infty, u_{r_1}^{(2)}) \cup (0, u_{r_2}^{(2)})$. The values of f at $u_{r_2}^{(2)}$ and $u_{r_1}^{(2)}$ are

$$f(u_{r_2}^{(2)}) = \left(\left(\frac{(K-\kappa)(1-\eta)(2\eta-1)^2}{\epsilon^2} \right)^{\frac{1}{3}} + (1-\eta) \right)^2 \times \left((K-\kappa)^{\frac{1}{3}} \left(\frac{\epsilon^2(2\eta-1)}{(1-\eta)} \right)^{\frac{2}{3}} + \epsilon^2 \right),$$

$$f(u_{r_1}^{(2)}) = \left(\left(\frac{(K-\kappa)(1-\eta)}{\epsilon^2} \right)^{\frac{1}{3}} + (1-\eta) \right)^2 \times \left((K-\kappa)^{\frac{1}{3}} \left(\frac{\epsilon^2}{(1-\eta)} \right)^{\frac{2}{3}} + \epsilon^2 \right).$$

Consequently, the minimum of f is obtained at $u^\star := u_{r_2}^{(2)}$. By (33) and (34), the means $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$ and the covariance matrices $\boldsymbol{\Sigma}_1$, $\boldsymbol{\Sigma}_2$ satisfy: $\mathbf{0}_K = u^\star \boldsymbol{\mu}_2 + (1-u^\star)\boldsymbol{\mu}_1$, $\boldsymbol{\Sigma}_1 = \text{diag}(\mathbf{1}_\kappa; \mathbf{0}_{K-\kappa})$ and $\boldsymbol{\Sigma}_2 = \text{diag}(\mathbf{1}_\kappa; (u^\star)^{-2}\mathbf{1}_{K-\kappa})$. Moreover, the norms of $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ can be computed from (34) as $u^\star\epsilon$ and $(1-u^\star)\epsilon$, respectively.

E.5 Proof of Proposition 3.3

Notice that since $\boldsymbol{\Sigma}_0 \in \mathcal{S}_{++}^K$, $\det(\boldsymbol{\Sigma}_0) > 0$. On the other hand, since $\boldsymbol{\Sigma}_1 \in \mathcal{S}_+^K$ with $\text{rank}(\boldsymbol{\Sigma}_1) = \kappa < K$, $\det(\boldsymbol{\Sigma}_1) = 0$. Therefore,

$$\log \frac{\det(\boldsymbol{\Sigma}_0)}{\det(\boldsymbol{\Sigma}_1)} = \log\det(\boldsymbol{\Sigma}_0) - \log\det(\boldsymbol{\Sigma}_1) = \infty.$$

This and (25) imply that $KL(\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) || \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)) = \infty$.

F ADDITIONAL DETAILS ON THE BENCHMARK METHODS

We overview the benchmark methods compared with MAW, where we present them according to alphabetical order of names. We will include all tested codes in a supplemental webpage.

For completeness, we mention the following links (or papers with links) we used for the different codes. For DSEBMs and DAGMM we used the codes of (Golan & El-Yaniv, 2018). For LOF, OCSVM and IF we used the scikit-learn (Buitinck et al., 2013) packages for novelty detection. For OCGAN we used its TensorFlow implementation from `https://pypi.org/project/ocgan`. For RSRAE, we adapted the code of (Lai et al., 2020) to novelty detection.

All experiments were executed on a Linux machine with 64GB RAM and four GTX1080Ti GPUs.

We remark that for the neural networks based methods (DAGMM, DSEBMs, OCGAN and RSRAE), we followed similar implementation details as the one described in §A.3 for MAW.

Deep Autoencoding GMM (DAGMM) (Zong et al., 2018): This method uses a deep autoencoder model. It optimizes an end-to-end structure that contains both an autoencoder and an estimator for a GMM. Anomalies are detected using this GMM. We remark that this mixture model is proposed for the inliers. An improved version of DAGMM was recently proposed in (Fan et al., 2020).

Deep Structured Energy-Based Models (DSEBMs) (Zhai et al., 2016): Its decision is based on an energy function which is the negative log probability that a sample follows the data distribution. An autoencoder is used for the energy-based model in order to avoid the need of complex sampling.

Isolation Forest (IF) (Liu et al., 2008): It iteratively constructs special binary trees for the training set and identifies anomalies in the test set as the ones with shortest average path lengths.

Local Outlier Factor (LOF) (Breunig et al., 2000): It measures the isolation of a data point from its surrounding neighbors by estimating the local density of this point using its k nearest neighbors. In novelty detection, it identifies novelties according to low density regions learned from the training data.

One-class Novelty Detection Using GANs (OCGAN) (Perera et al., 2019): It is composed of four NNs: a denoising autoencoder, two adversarial discriminators, and a classifier. It adversarially encourages the autoencoder to learn only the inlier features.

One-Class SVM (OCSVM) (Heller et al., 2003): It estimates the margin of the training set and uses it as the decision boundary for the test set. It commonly utilizes a radial basis function kernel.

Robust Subspace Recovery Autoencoder (RSRAE) (Lai et al., 2020): It uses an autoencoder with a linear RSR layer and an $\ell_{2,1}$-based penalty. The RSR layer extracts features of inliers in the latent code while helping to reject outliers. The instances with higher reconstruction errors are viewed as outliers. RSRAE trains a model using the training data. We then apply this model for detecting novelties in the test data.

G ADDITIONAL DETAILS ON THE DIFFERENT DATASETS

Below we provide additional details on the six datasets used in our experiments. We remark that each dataset contains several clusters (3 for COVID-19, 10 for CIFAR-10, 11 largest ones for Caltech101, 10 for Fashion MNIST, 2 for KDDCUP-99 and 5 for Reuters-21578,). Table 2 lists for each dataset (for both training and testing) the data types, numbers of clusters, dimensions, numbers of instances and numbers of inliers and outliers.

Table 2: Summary of properties of the datasets.

	Dataset information				Training		Testing	
Datasets	Type	#Clusters	Dimension	#Instances	#Inliers	#Outliers	#Inliers	#Outliers
COVID-19 (Radiography)	Image	3	$64 \times 64 \times 3$	15,161	160	$160 \times c$	60	$60 \times c_{\text{test}}$
CIFAR-10	Image	10	$32 \times 32 \times 3$	60,000	450	$450 \times c$	150	$150 \times c_{\text{test}}$
Caltech101	Image	11	$32 \times 32 \times 3$	9,146	100	$100 \times c$	100	$100 \times c_{\text{test}}$
Fashion MNIST	Image	10	$28 \times 28 \times 1$	70,000	300	$300 \times c$	60	$60 \times c_{\text{test}}$
KDDCUP-99	Feature	2	120	494,021	6000	$6000 \times c$	1200	$1200 \times c_{\text{test}}$
Reuters-21578	Feature	5	$26,147$	21,578	350	$350 \times c$	140	$140 \times c_{\text{test}}$

COVID-19 (Radiography) (Chowdhury et al., 2020): It contains chest X-ray images (RGB) labeled according to these categories: COVID-19 positive, normal and bacterial Pneumonia cases. We resize the images to size 64×64 and rescale the pixel intensities to lie in $[-1, 1]$. It is publicly available in `https://www.kaggle.com/tawsifurrahman/covid19-radiography-database`.

CIFAR-10 (Krizhevsky, 2009): It contains 10 categories of 32×32 RGB images of transportation vehicles and animals. We rescale the pixel intensities to lie in $[0, 1]$. The dataset is publicly available in `https://www.cs.toronto.edu/~kriz/cifar.html`.

Caltech101 (Fei-Fei et al., 2004): It contains RGB images of objects from 101 categories with identifying labels. Following Lai et al. (2020), we use the largest 11 classes and preprocess their images to have size 32×32 and rescale the pixel intensities to lie in $[-1, 1]$. It is publicly available in `http://www.vision.caltech.edu/Image_Datasets/Caltech101`.

Fashion MNIST (Xiao et al., 2017): It is an image dataset containing 10 categories of 28×28 grayscale images of clothing and accessories items. We rescaled the pixel intensities to lie in $[-1, 1]$. We obtained the dataset from the Keras dataset library `https://keras.io/api/datasets/fashion_mnist`.

KDDCUP-99 (Dua & Graff, 2017): It is a classic dataset for intrusion detection. It contains feature vectors of connections between internet protocols and a binary label for each feature vector identifying normal vs. abnormal ones. The abnormal ones are associated with an "attack" or "intrusion". The dataset is publicly available in `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`.

Reuters-21578 (Lewis, 1997): It contains 21,578 documents with 90 text categories having multi-labels. Following Lai et al. (2020), we consider the five largest classes with single labels. We utilize the scikit-learn packages: TFIDF and HashingVectorizer (Rajaraman & Ullman, 2011) to preprocess the documents into 26,147 dimensional vectors. It is publicly available in `https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection`.

According to the above description, the numbers of clusters of these datasets are 3, 10, 11, 10, 2 and 5, respectively. We

remark that COVID-19, CIFAR-10, Caltech101, Fashion MNIST and Reuters-21578 separate between training and testing data points. For KDDCUP-99, we randomly split it into training and testing datasets of equal sizes.

H NUMERICAL RESULTS OF THE EXPERIMENTS

We present as tables the numerical values depicted in Figs. 2 and 3. Tables 3-14 report the averaged AUC and AP scores with training outliers/inliers ratio $c \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ that were depicted in Fig. 2. Each table describes one of the averaged scores (AUC or AP) for one of the six datasets (COVID-19, CIFAR-10, Caltech101, Fashion MNIST, KDDCUP-99 and Reuters-21578) and also indicates the standard deviation of each value. The outperforming methods are marked in bold.

Tables 15-18 record the averaged AUC and AP scores with training outliers/inliers ratio $c \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ that were depicted in Fig. 3. Each table describes one of the averaged scores (AUC or AP) for one of either KDDCUP-99 or COVID-19 and also indicates the standard deviation of each value. The outperforming methods are boldfaced.

Table 3: AUC scores of COVID-19.

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	**0.652** $\pm$ 0.021	**0.609** $\pm$ 0.018	**0.576** $\pm$ 0.019	0.531 $\pm$ 0.020	0.504 $\pm$ 0.010
DAGMM	0.527$\pm$ 0.068	0.545$\pm$ 0.051	0.518 $\pm$ 0.062	0.504 $\pm$ 0.060	0.503 $\pm$ 0.057
DSEBMs	0.451$\pm$ 0.000	0.451$\pm$ 0.000	0.451 $\pm$ 0.000	0.451 $\pm$ 0.000	0.451 $\pm$ 0.000
IF	0.574	0.541	0.515	0.493	0.469
LOF	0.642	0.588	0.542	**0.536**	**0.519**
OCGAN	0.472$\pm$ 0.000	0.472$\pm$ 0.000	0.465 $\pm$ 0.000	0.445 $\pm$ 0.000	0.431 $\pm$ 0.000
OCSVM	0.528	0.528	0.528	0.535	0.521
RSRAE	0.535$\pm$ 0.031	0.507$\pm$ 0.028	0.456 $\pm$ 0.023	0.434 $\pm$ 0.018	0.407 $\pm$ 0.011

Table 4: AP scores of COVID-19.

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	0.459 $\pm$ 0.014	**0.442** $\pm$ 0.011	**0.424** $\pm$ 0.018	0.368 $\pm$ 0.015	0.353 $\pm$ 0.013
DAGMM	0.354$\pm$ 0.053	0.390 $\pm$ 0.057	0.316 $\pm$ 0.052	0.357 $\pm$ 0.050	0.348 $\pm$ 0.047
DSEBMs	0.372$\pm$ 0.000	0.375 $\pm$ 0.000	0.364 $\pm$ 0.000	0.360 $\pm$ 0.000	0.358 $\pm$ 0.000
IF	0.425	0.404	0.392	0.373	0.363
LOF	**0.463**	0.422	0.402	**0.374**	**0.371**
OCGAN	0.381$\pm$ 0.000	0.381 $\pm$ 0.000	0.381 $\pm$ 0.000	0.373 $\pm$ 0.000	0.350 $\pm$ 0.000
OCSVM	0.315	0.315	0.315	0.372	0.365
RSRAE	0.388$\pm$ 0.018	0.377 $\pm$ 0.016	0.355 $\pm$ 0.011	0.352 $\pm$ 0.010	0.340 $\pm$ 0.009

Table 5: AUC scores of CIFAR-10.

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	0.621 $\pm$ 0.013	**0.609** $\pm$ 0.014	**0.607** $\pm$ 0.012	0.600 $\pm$ 0.010	**0.595** $\pm$ 0.013
LOF	0.582	0.574	0.559	0.551	0.539
OCSVM	0.595	0.587	0.580	0.564	0.570
IF	0.603	0.586	0.596	0.581	0.569
RSRAE	**0.638** $\pm$ 0.010	0.607 $\pm$ 0.017	0.599 $\pm$ 0.023	**0.610** $\pm$ 0.025	0.589 $\pm$ 0.023
DSEBMs	0.586 $\pm$ 0.006	0.584 $\pm$0.006	0.580 $\pm$ 0.004	0.576$\pm$ 0.006	0.556 $\pm$0.006
OCGAN	0.501 $\pm$0	0.501 $\pm$0	0.499 $\pm$0	0.487 $\pm$ 0	0.476 $\pm$0
DAGMM	0.574 $\pm$ 0.030	0.557 $\pm$ 0.035	0.541 $\pm$0.037	0.510 $\pm$ 0.0331	0.545 $\pm$ 0.037

Table 6: AP scores of CIFAR-10.

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	0.427 ± 0.010	**0.419** ±0.012	0.414 ± 0.011	**0.400**±0.009	**0.411** ±0.011
LOF	0.395	0.036	0.377	0.374	0.371
OCSVM	0.408	0.400	0.393	0.378	0.385
IF	0.416	0.395	0.403	0.389	0.373
RSRAE	**0.434** ± 0.011	0.412 ±0.020	**0.417** ± 0.022	0.391 ± 0.019	0.400 ± 0.014
DSEBMs	0.391 ±0.008	0.388 ± 0.008	0.386 ±0.004	0.382 ± 0.006	0.379±0.003
OCGAN	0.342 ± 0	0.340±0	0.339 ±0	0.337 ± 0	0.335±0
DAGMM	0.378 ±0.049	0.369 ±0.041	0.355 ± 0.030	0.308 ± 0.026	0.352 ± 0.047

Table 7: AUC scores of Caltech101.

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	**0.801** ± 0.017	**0.760** ± 0.028	**0.700** ± 0.038	**0.608** ± 0.031	**0.570** ± 0.021
DAGMM	0.684 ± 0.100	0.588 ± 0.115	0.500± 0.100	0.509 ± 0.101	0.514 ± 0.095
DSEBMs	0.536 ± 0.011	0.612± 0.025	0.577 ± 0.030	0.564 ± 0.021	0.536 ± 0.021
IF	0.755	0.694	0.626	0.575	0.540
LOF	0.674	0.593	0.495	0.436	0.411
OCGAN	0.494 ± 0.000	0.494± 0.000	0.494± 0.000	0.500 ± 0.000	0.500 ± 0.000
OCSVM	0.682	0.618	0.577	0.538	0.516
RSRAE	0.774 ± 0.027	0.722 ± 0.041	0.664 ± 0.082	0.579 ± 0.047	0.568 ± 0.036

Table 8: AP scores of Caltech101.

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	**0.634** ± 0.027	**0.572** ± 0.039	**0.531** ± 0.064	0.412 ± 0.029	0.414 ± 0.021
DAGMM	0.574± 0.088	0.422 ± 0.112	0.308 ± 0.102	0.351 ± 0.074	0.363 ± 0.076
DSEBMs	0.385± 0.003	0.472± 0.051	0.398±0.019	0.383 ± 0.023	0.365 ± 0.028
IF	0.545	0.486	0.430	0.304	0.371
LOF	0.460	0.400	0.337	0.304	0.290
OCGAN	0.362± 0.000	0.362± 0.000	0.362 ± 0.000	0.362 ± 0.000	0.362 ± 0.000
OCSVM	0.472	0.419	0.380	0.352	0.339
RSRAE	0.595± 0.038	0.551 ± 0.045	0.495 ±0.073	**0.425** ± 0.040	**0.443** ± 0.027

Table 9: AUC scores of Fashion MNIST

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	**0.897** ± 0.013	**0.879** ± 0.011	**0.852** ± 0.022	0.830 ± 0.017	0.801 ± 0.016
DAGMM	0.607 ± 0.093	0.376 ± 0.070	0.427 ± 0.090	0.401 ± 0.078	0.411 ± 0.081
DSEBMs	0.730 ± 0.092	0.729 ± 0.105	0.739 ± 0.086	0.723 ± 0.106	0.687 ± 0.096
IF	0.893	0.875	0.843	**0.834**	**0.827**
LOF	0.569	0.507	0.476	0.468	0.458
OCGAN	0.542 ± 0.006	0.538 ± 0.004	0.544 ± 0.014	0.531 ± 0.003	0.525 ± 0.004
OCSVM	0.895	0.874	0.848	0.831	0.814
RSRAE	0.860 ± 0.022	0.848 ± 0.022	0.829 ± 0.042	0.831 ± 0.028	0.808 ± 0.028

Table 10: AP scores of Fashion MNIST

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	**0.788** ±0.013	0.754 ± 0.014	0.723±0.029	0.686 ± 0.025	0.672 ±0.021
DAGMM	0.482 ±0.051	0.303 ±0.057	0.334 ±0.113	0.318 ±0.056	0.330 ± 0.038
DSEBMs	0.600 ± 0.045	0.609± 0.120	0.613±0.089	0.605 ±0.086	0.565 ± 0.072
IF	0.768	0.724	0.693	0.665	0.642
LOF	0.382	0.331	0.308	0.301	0.294
OCGAN	0.504 ± 0.002	0.503 ± 0.003	0.500 ± 0.059	0.495 ± 0.001	0.493 ± 0.001
OCSVM	0.801	**0.768**	**0.735**	**0.696**	0.664
RSRAE	0.749 ± 0.029	0.736 ± 0.032	0.716 ± 0.048	0.683 ± 0.036	**0.680** ± 0.042

Table 11: AUC scores of KDDCUP-99.

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	**0.945** ± 0.028	**0.906** ± 0.018	**0.832** ± 0.016	**0.775** ± 0.023	**0.731** ± 0.017
DAGMM	0.614 ± 0.083	0.660 ± 0.109	0.584 ± 0.133	0.457 ± 0.099	0.521 ± 0.089
DSEBMs	0.514± 0.000	0.499 ± 0.000	0.497 ± 0.000	0.496 ± 0.000	0.496 ± 0.000
IF	0.811	0.850	0.807	0.750	0.706
LOF	0.480	0.527	0.516	0.527	0.530
OCGAN	0.651 ± 0.157	0.552 ± 0.157	0.617 ± 0.191	0.517 ± 0.146	0.628 ± 0.155
OCSVM	0.502	0.568	0.567	0.555	0.534
RSRAE	0.815 ± 0.031	0.839 ± 0.059	0.774 ± 0.086	0.735 ± 0.066	0.710 ± 0.056

Table 12: AP scores of KDDCUP-99.

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	**0.765**± 0.025	**0.732** ± 0.015	**0.647** ± 0.012	**0.594** ± 0.014	0.556 ± 0.014
DAGMM	0.446 ± 0.047	0.506 ± 0.064	0.459 ± 0.087	0.373 ± 0.109	0.464 ± 0.998
DSEBMs	0.450 ± 0.000	0.447 ± 0.000	0.446 ± 0.000	0.444 ± 0.000	0.444 ± 0.000
IF	0.636	0.6331	0.562	0.493	0.457
LOF	0.391	0.407	0.392	0.394	0.391
OCGAN	0.582 ± 0.132	0.472 ± 0.163	0.525 ± 0.133	0.418 ± 0.136	0.535 ± 0.133
OCSVM	0.543	0.598	0.595	0.438	0.426
RSRAE	0.704 ± 0.048	0.698 ± 0.050	0.606 ± 0.065	0.584 ± 0.034	**0.574** ± 0.046

Table 13: AUC scores of Reuters-21578.

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	0.885 ± 0.028	**0.830** ± 0.013	0.770 ± 0.017	**0.700** ± 0.002	**0.648** ± 0.016
DAGMM	0.500 ± 0.000	0.511 ± 0.027	0.566 ± 0.110	0.559 ± 0.087	0.570 ± 0.091
DSEBMs	**0.887** ± 0.012	0.825 ± 0.012	**0.790** ± 0.015	0.690 ± 0.002	0.648 ± 0.010
IF	0.544	0.535	0.520	0.453	0.452
LOF	0.757	0.612	0.579	0.631	0.616
OCGAN	0.648 ± 0.127	0.477 ± 0.129	0.498 ± 0.140	0.519 ± 0.132	0.502 ± 0.099
OCSVM	0.882	0.817	0.785	0.673	0.640
RSRAE	0.786 ± 0.042	0.755 ± 0.034	0.716 ± 0.033	0.605 ± 0.001	0.494 ± 0.004

Table 14: AP scores of Reuters-21578.

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	0.755 ± 0.041	0.677 ± 0.026	0.627 ± 0.029	**0.518** ± 0.004	0.474 ± 0.013
DAGMM	0.316 ± 0.000	0.316 ± 0.013	0.365 ± 0.020	0.362 ± 0.015	0.372 ± 0.012
DSEBMs	**0.763** ± 0.012	**0.697** ± 0.011	**0.666** ± 0.007	0.515 ± 0.003	0.473 ± 0.003
IF	0.368	0.372	0.365	0.301	0.298
LOF	0.580	0.438	0.421	0.498	**0.486**
OCGAN	0.408 ± 0.045	0.334 ± 0.098	0.365 ± 0.106	0.504 ± 0.083	0.497 ± 0.094
OCSVM	0.746	0.681	0.637	0.467	0.438
RSRAE	0.593 ± 0.051	0.563 ± 0.035	0.488 ± 0.036	0.403± 0.001	0.415 ± 0.003

Table 15: AUC scores of KDD-99 for variations of MAW

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	**0.945** ± 0.028	**0.906** ± 0.018	**0.832** ± 0.016	**0.775** ± 0.023	**0.731** ± 0.017
MAW-MSE	0.844 ± 0.039	0.812 ± 0.032	0.746 ± 0.044	0.709 ± 0.020	0.675 ± 0.014
MAW-KL divergence	0.905 ± 0.026	0.863 ± 0.028	0.801 ± 0.029	0.752 ± 0.016	0.696 ± 0.018
MAW-same rank	0.912 ± 0.023	0.868 ± 0.011	0.797 ± 0.022	0.750 ± 0.012	0.699 ± 0.040
MAW-single Gaussian	0.914 ± 0.016	0.862 ± 0.021	0.796 ± 0.013	0.751 ± 0.040	0.701 ± 0.045
MAW-diagonal cov.	0.918 ± 0.023	0.858 ± 0.020	0.801 ± 0.044	0.743 ± 0.017	0.703 ± 0.015
VAE	0.821 ± 0.048	0.785 ± 0.027	0.732 ± 0.046	0.717 ± 0.018	0.685 ± 0.027
GMM prior	0.677 ± 0.009	0.635±0.013	0.604 ± 0.006	0.562 ± 0.015	0.517±0.014

Table 16: AP scores of KDDCUP-99 for variations of MAW

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	**0.765** ± 0.025	**0.732** ± 0.015	**0.647** ± 0.012	**0.594**± 0.014	**0.556** ± 0.014
MAW-MSE	0.715 ± 0.079	0.589 ± 0.058	0.524 ± 0.053	0.463±0.042	0.410 ± 0.028
MAW-KL divergence	0.735 ± 0.028	0.676 ± 0.028	0.618 ± 0.024	0.579±0.023	0.509±0.017
MAW-same rank	0.725 ± 0.028	0.681 ±0.015	0.622 ± 0.024	0.572 ±0.017	0.532 ±0.038
MAW-single Gaussian	0.737 ± 0.018	0.675 ±0.023	0.620 ± 0.025	0.569±0.036	0.519 ±0.044
MAW-diagonal cov.	0.724 ± 0.021	0.678 ± 0.035	0.589 ± 0.064	0.546 ±0.019	0.512 ±0.016
VAE	0.642 ± 0.030	0.555 ± 0.043	0.524 ± 0.028	0.478±0.024	0.450±0.015
GMM prior	0.604 ± 0.010	0.587±0.013	0.557 ± 0.009	0.534 ± 0.011	0.501±0.015

Table 17: AUC scores of COVID-19 for variations of MAW

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	**0.652** ± 0.021	**0.609** ± 0.018	**0.576** ± 0.019	**0.531** ±0.020	**0.504**±0.010
MAW-MSE	0.602 ± 0.022	0.554 ±0.063	0.528 ± 0.041	0.507± 0.014	0.479±0.021
MAW-KL divergence	0.614 ± 0.025	0.580 ± 0.026	0.508 ± 0.064	0.476 ± 0.023	0.463±0.016
MAW-same rank	0.604 ± 0.031	0.574 ± 0.048	0.527 ± 0.044	0.430 ± 0.017	0.408±0.021
MAW-single Gaussian	0.621 ± 0.027	0.586±0.029	0.507 ± 0.047	0.492±0.021	0.472±0.019
MAW-diagonal cov.	0.600 ± 0.029	0.586± 0.030	0.535 ± 0.035	0.446 ±0.028	0.439 ±0.038
VAE	0.619 ± 0.073	0.565±0.065	0.522 ± 0.049	0.508 ± 0.023	0.473±0.016
GMM prior	0.548 ± 0.012	0.514±0.008	0.489 ± 0.010	0.476 ± 0.011	0.469±0.009

Table 18: AP scores of COVID-19 for variations of MAW

	Training ratio of outliers per inliers, c				
Methods	0.1	0.2	0.3	0.4	0.5
MAW	**0.459** ± 0.014	**0.442** ± 0.011	**0.424** ± 0.018	**0.368**±0.015	**0.353**±0.013
MAW-MSE	0.421 ± 0.015	0.395 ± 0.025	0.377 ± 0.012	0.332±0.013	0.328 ± 0.020
MAW-KL divergence	0.427 ± 0.016	0.403± 0.012	0.370 ± 0.021	0.322±0.017	0.313 ±0.013
MAW-same rank	0.422 ± 0.021	0.413 ±0.026	0.375 ± 0.019	0.344 ±0.023	0.335 ±0.017
MAW-single Gaussian	0.425± 0.019	0.409 ± 0.012	0.374 ± 0.016	0.339± 0.014	0.329±0.016
MAW-diagonal cov.	0.412 ± 0.016	0.397 ± 0.018	0.369 ± 0.012	0.343±0.009	0.330 ±0.009
VAE	0.412 ± 0.030	0.411 ± 0.043	0.379 ± 0.028	0.341±0.011	0.333±0.013
GMM prior	0.389 ± 0.009	0.382±0.006	0.362 ± 0.005	0.346 ± 0.009	0.309 ±0.0096