Multivariate Triangular Quantile Maps for Novelty Detection

by

Jingjing Wang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2024

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This thesis includes material that was presented at NeurIPS [1]. I proposed the ideas, conducted most of the experiments, and wrote the first draft. My supervisor, Professor Yaoliang Yu, contributed ideas, provided feedback, and improved my draft. The other co-author, Sun Sun, also helped to run experiments and polished the paper.

Abstract

Novelty detection, a fundamental task in the field of machine learning, has drawn a lot of recent attention due to its wide-ranging applications and the rise of neural approaches. In this thesis, we present a general framework for neural novelty detection that centers around a multivariate extension of the univariate quantile function. Our general framework unifies and extends many classical and recent novelty detection algorithms, and opens the way to exploit recent advances in flow-based neural density estimation. We adapt the multiple gradient descent algorithm to obtain the first efficient end-to-end implementation of our framework that is free of tuning hyperparameters. Extensive experiments over a number of synthetic and real datasets confirm the efficacy of our proposed method against state-of-the-art alternatives.

Acknowledgements

I would like to thank all the people who made this thesis possible.

Dedication

To my family.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Novelty detection, the fundamental task in machine learning that detects "novel" or "unusual" samples in a data stream. It has wide-ranging applications such as network intrusion detection [2], medical signal processing [3], jet design [4], video surveillance [5, 6], image scene analysis [7, 8], document classification [9, 10], reinforcement learning [11], etc.; see the review articles [12, 13, 14, 15] for more insightful applications.

Over the last two decades or so, many novelty detection algorithms have been proposed and studied in the field of machine learning, of which the statistical approach that aims to identify low-density regions of the underlying data distribution has been the most popular [e.g. 16, 17, 18, 19]. More recently, new novelty detection algorithms based on deep neural networks [e.g. 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33] have drawn a lot of attention as they significantly improve upon their non-neural counterparts (i.e., shallow methods), especially in domains such as image and video where complex high-dimensional structures abound.

1.1 Novelty Detection as One-Class Classification

Novelty detection methods [34] are usually categorized into supervised, semi-supervised, and unsupervised approaches. In the supervised setting, novelty detection is seen as binary classification to learn the decision boundary from labeled nominal and novel instances in the training dataset. However, due to the unpredictability and difficulty in acquiring novel samples and accurately labeling training data, unsupervised or semi-supervised algorithms have become increasingly popular and more widely used than their supervised alternatives. On the other hand, the task, if categorized as unsupervised approach, becomes particularly challenging without the utilization of

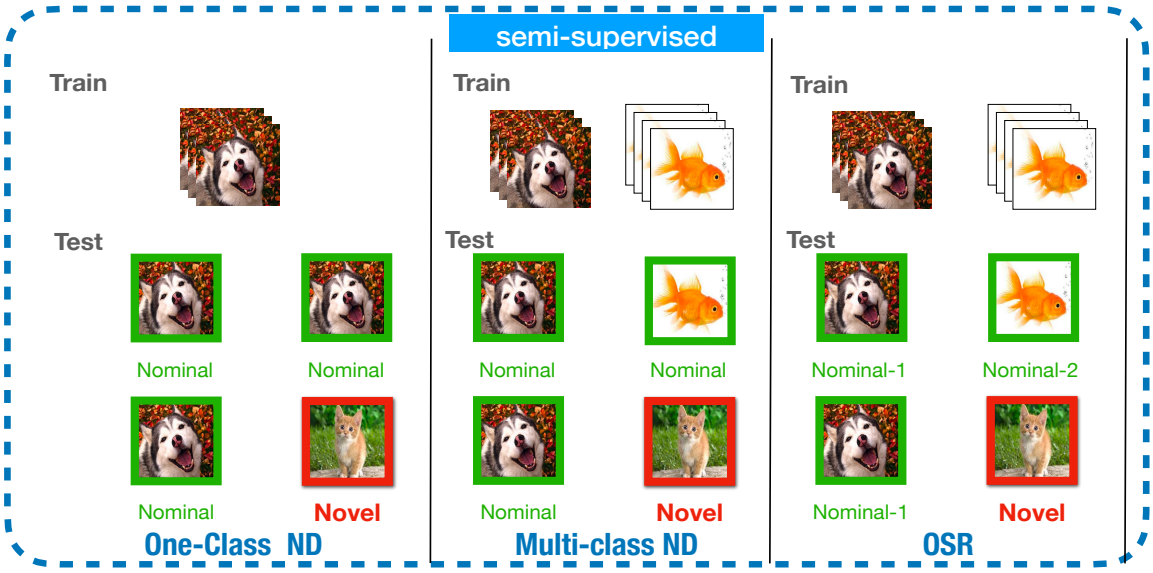

Figure 1.1: Semi-supervised Novelty Detection

any potential label information. Therefore, most novelty detection approaches are in the semi-supervised setting, in which the training dataset usually includes nominal samples, sometimes also includes partially labeled novel samples. In Figure 1.1, we list three popular semi-supervised settings for novelty detection: one-class novelty detection, multi-class novelty detection, and open set recognition (OSR).

In this thesis, we only focus on one-class novelty detection. Novelty detection is seen as "one-class classification"[15], in which a model is constructed to describe nominal training data. The novelty approach is typically used when the quantity of novel data is insufficient to construct explicit models for novel classes and the quantity of nominal data is very large such that the model of nominal may be accurate.

1.2 Formulation

We follow the standard setup for novelty detection [e.g. 14]: Given n i.i.d. samples $\wr \mathbf{X}_1, \ldots, \mathbf{X}_n \int$ from an unknown distribution P over $\mathbb{R}^d$, we want to decide if a new sample $\tilde{\mathbf{X}}$ is "novel", i.e., if it is unlikely to come from the same distribution P.

Due to lack of supervision, the notion of "novelty" is not well-defined. Practically, a popular surrogate is to identify the low-density regions of the distribution P [16, 19, 17], as samples from

these areas are probabilistically unlikely. For simplicity we assume the underlying distribution P has a density p w.r.t. the Lebesgue measure.

1.3 Outline

The current dominant approach in novelty detection is the use of the likelihood or other metrics from a generative model trained on the in-distribution data. This thesis offers a closer look at these recent neural novelty detection algorithms, by making a connection to recent flow-based generative modelling techniques [35].

In §2 we show that the triangular map studied in [35] for neural density estimation serves as a natural extension of the classical univariate quantile function to the multivariate setting. Since density estimation is extremely challenging in high dimensions, recent neural novelty detection algorithms all extract a lower dimensional latent representation, whose probabilistic properties can then be captured by our multivariate triangular quantile maps. Based on this observation we propose a general framework for neural novelty detection that includes as special cases many classical approaches such as one-class SVM [17] and support vector data description [18], as well as many recent neural approaches [e.g. 31, 24, 26, 22, 21]. This unified view of neural novelty detection enables us to better understand the similarities and subtle differences of the many existing approaches, and provides some guidance on designing next-generation novelty detection algorithms.

More importantly, our general framework makes it possible to effortlessly plug in recent flow-based neural density estimators, which have been shown to be surprisingly effective even in moderately high dimensions. Furthermore, centering our framework around the (multivariate) triangular quantile map (TQM) also enables us to unify the two scoring strategies in the literature [36]: we can either threshold the density function [16, 19] or the (univariate) quantile function [17, 18]. Using the multivariate triangular quantile map, for the first time we can simultaneously perform both, without incurring any additional cost.

In §3, motivated by the sub-optimality of pre-training we cast our novelty detection framework as multi-objective optimization [37] and apply the multiple gradient descent algorithm [38, 39, 40] for the first time. We present an efficient implementation that learns the TQM consistently, end-to-end and free of tuning hyperparameters. In §4 we perform extensive experiments on a variety of datasets and verify the effectiveness of our framework against state-of-the-art alternatives.

1.4 Contributions

Finally, we end this chapter by summarizing the contributions made by this thesis.

- We extend the univariate quantile function to the multivariate setting through increasing triangular maps. This multivariate triangular quantile map may be of independent interest for many other problems involving multivariate probabilistic modelling.
- We present a new framework for neural novelty detection, which unifies and extends many existing approaches including the celebrated one-class SVM and many recent neural ones.
- For the first time we apply the multiple gradient descent algorithm to novelty detection and obtain an efficient end-to-end implementation of our framework that is free of any tuning hyperparameters.
- We perform extensive experiments to compare to existing novelty detection baselines and to confirm the efficacy of our proposed framework.

Our PyTorch implementation is available at https://github.com/GinGinWang/MTQ. This thesis is an extended version of my published work [1] at NeurIPS.

Chapter 2

General Framework for Novelty Detection

In this chapter, we present a general framework for novelty detection centered around Triangular Quantile Maps (TQM). Our framework builds on recent progresses in generative modelling and unifies and extends many existing works.

First, we extend the univariate quantile to multivariate setting using the TQM, and also give a brief introduction of multivariate quantiles which lack a unified definition. We introduce our multivariate generalization of the univariate quantile function in §2.1 and propose a novel multivariate quantile function inspired by the transformation interpretation of univariate quantiles. Additionally, we explore other multivariate quantile definitions and conduct a comparative analysis with our proposed approach.

2.1 Multivariate Generalization of the Quantile Function

Before delving into our general framework, which allows for simultaneous establishment of density thresholding rules and quantile thresholding rules without any additional cost, let's first explore the multivariate generalization of the quantile function.

2.1.1 Transformation interpretation of the Quantile Function

Recall that the cumulative distribution function (CDF) F and the quantile function Q of a *univariate* random variable X is defined as:

$$F(x) = \Pr(X \leq x), \qquad Q(u) = F^{-1}(u) := \inf\{x : F(x) \geq u\}. \tag{2.1}$$

While the CDF can be easily generalized to the multivariate setting, it is not so obvious for the quantile function, as its definition intrinsically relies on the total ordering on the real line. However, following [e.g. 41, 42] we observe that if U follows the uniform distribution over the interval $[0, 1]$, then $Q(U)$ follows the distribution F. In other words, *the quantile function can be defined as a mapping that pushes the uniform distribution over* $[0, 1]$ *into the distribution* F *of interest*. This mapping is fundamental in statistical analysis and is essential for various applications such as generating random variates from the desired distribution F. More importantly, This alternative interpretation allows us to extend the quantile function to the multivariate setting.

Now the question is whether the multivariate quantile function can be depicted as a transformation that converts the uniform distribution over $[0, 1]^n$ into the desired distribution F over $\mathbb{R}^d$ of interest. The following are the prerequisites that such a mapping satisfy.

- **Existence and Uniqueness**: For every point in the uniform distribution over $[0, 1]^n$, there exists a unique corresponding point in the distribution F, and vice versa. Mathematically, the mapping should be bijective.
- **Preservation of Order**: If one point in the uniform distribution is greater than another, its corresponding point in F should also be greater than the corresponding point of the other point in F.
- **Efficiency and Computational Feasibility**: The mapping should be computationally feasible and efficient, especially for high-dimensional data. This ensures that the transformation can be practically applied in various statistical analyses and simulations.
- **Expressive Power**: The implementation of the mapping must have complicated structure to model any desired distribution and can be easily adjusted the complicated level considering the trade-off with computational efficiency.

In the upcoming section, we will introduce our novel multivariate quantile function designed to fulfill all the aforementioned requirements.

2.1.2 Triangular Quantile Map

We recall that a mapping $\mathbf{T} = (T_1, \ldots, T_d) : \mathbb{R}^d \to \mathbb{R}^d$ is called *triangular* if for all $j = 1, \ldots, d$, the j-th component T_j depends only on the first j coordinates of the input, and it is called *increasing* if for all j, T_j is increasing w.r.t. the j-th coordinate when all other coordinates are fixed. We call $\mathbf{T}$ triangular since its derivative is always a triangular matrix (and vice versa). In Figure 2.1, we provide a concise illustration demonstrating the definition of increasing triangular maps.

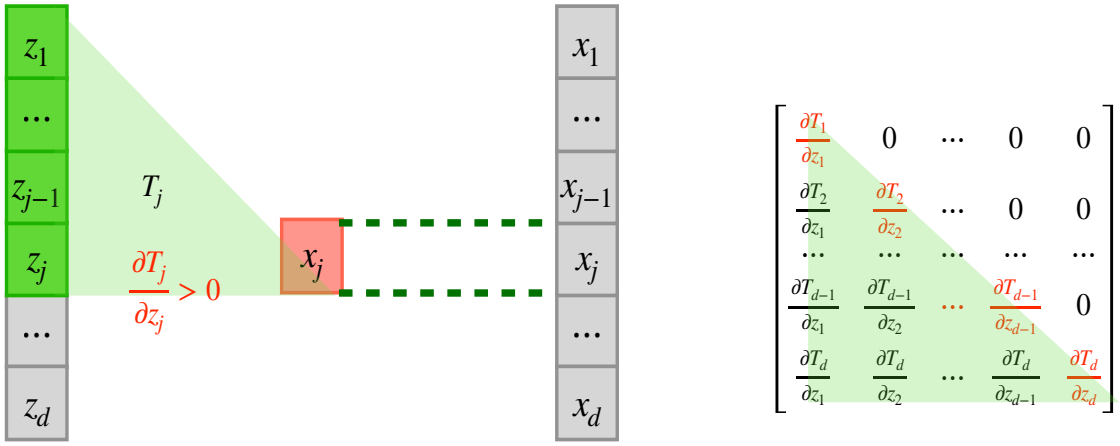

Figure 2.1: Illustration of the definition of increasing triangular maps.

Definition 1 (Triangular Quantile Map (TQM)) *Let* $\mathbf{X}$ *be a random vector in* $\mathbb{R}^d$*, and let* $\mathbf{U}$ *be uniform over the unit hypercube* $[0,1]^d$*. We call an* ***increasing triangular*** *map* $\mathbf{Q} = \mathbf{Q}_\mathbf{X} : [0,1]^d \to \mathbb{R}^d$ *the triangular quantile map of* $\mathbf{X}$ *if* $\mathbf{Q}(\mathbf{U}) \sim \mathbf{X}$*, where* $\sim$ *means equality in distribution.*

Here we give two special cases to illustrate the definition of the TQM and the second one will be frequently used later when stacking TQMs to remove constraint on the support of TQMs.

Remark 1 (Reduced to Univariate) $\mathbf{Q}$ *is the quantile function when* $d = 1$*, i.e., the TQM is reduced to the univariate quantile function. .*

Remark 2 (Inverse Gaussian CDF) $\mathbf{Q} = \mathbf{\Phi}^{-1}$*, where* $\mathbf{\Phi} = (\Phi, \ldots, \Phi)$ *with* Φ *the CDF of standard univariate Gaussian.*

Note that the TQM $\mathbf{Q}$ is *vector-valued*, unlike the CDF which is always real-valued. The *existence* and *uniqueness* of $\mathbf{Q}$ follows from results in [43]. Our definition immediately leads to the following quantile change-of-variable formula (cf. the usual change-of-variable formula for densities):

Proposition 1 (Quantile Change-of-Variable Formula) *Let* $\mathbf{T} : \mathbb{R}^d \to \mathbb{R}^d$ *be an increasing triangular map. If* $\mathbf{Y} = \mathbf{T}(\mathbf{X})$*, then*

$$\mathbf{Q}_\mathbf{Y} = \mathbf{T} \circ \mathbf{Q}_\mathbf{X}. \tag{2.2}$$

Practically, eq. (2.2) allows us to easily stack elementary parameterizations of increasing triangular maps together and still obtain a valid TQM. This stacking property is very useful when designing and adjusting the complexity level of the TQM. For example, when the target distribution is complicated and we need a higher complexity of the mapping model, we can stack a few more TQMs (i.e., increasing the number of parameters of the designed model), as shown in Figure 2.2.

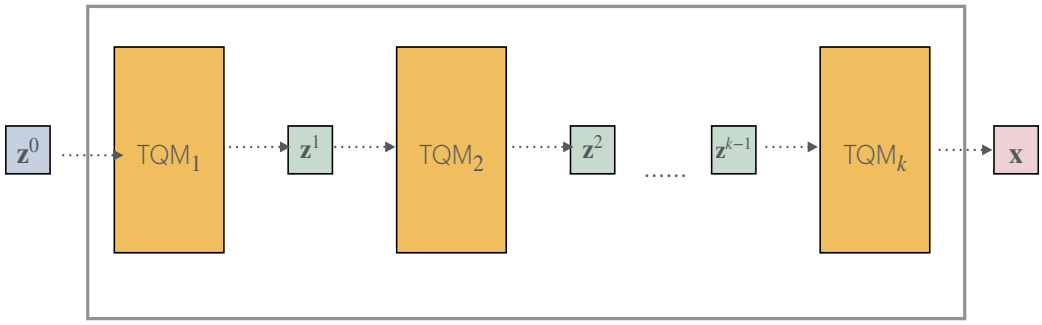

Figure 2.2: Schematic of a valid TQM by stacking multiple TQMs.

Overall, following Definition 1, we know that the TQM satisfy the existence and uniqueness, and preservation of order. Also, from Proposition 1, we can control the expressive power or model complexity by stacking. The only unmentioned prerequisite is efficiency, we will go to details into how to implement computationally convenient TQMs in §3.

2.1.3 Other Multivariate Quantile Function

To our best knowledge, a similar definition, through conditional univariate quantiles, appeared in a number of works [44, 45, 46, 47], albeit mostly as a theoretical tool. Our definition makes the important triangular structure explicit and amenable to parameterization through deep networks. Needless to say, when $d = 1$, the triangular property is vacuous and our definition reduces to the classical quantile function. For a more comprehensive introduction to triangular maps and its recent rise in machine learning, see [35, 48, 49].

A different definition of the multivariate quantile map, based on the theory of optimal transport [50], is discussed in a number of recent works [e.g. 41, 42, 51]: $\mathbf{Q}$ is instead constrained to be maximally cyclically monotone, i.e. it is the subdifferential of some convex function. On one hand, this definition is invariant to permutations of the input coordinates while ours is not. On the other hand, our definition is composition friendly (see Proposition 1) hence can easily exploit

recent progresses in deep generative models, as we will see shortly. The two definitions coincide with each other only when reduced to the univariate case.

We note that the recent work of Inouye and Ravikumar [52] proposed yet another similar definition where $\mathbf{Q}$ (termed density destructor there) is only required to be invertible. However, this definition does not lead to a *unique* quantile map and it is less computationally convenient.

2.2 Framework for Novelty Detection based on the TQM

We are now ready to present our general framework centering around the TQM for novelty detection. Let $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^m$ be a feature map and $\mathbf{X}$ a random sample from the unknown density p. We propose to learn the density[1] $\mathbf{f}_{\#}p$ of the latent random vector $\mathbf{Z} = \mathbf{f}(\mathbf{X})$ using the approach illustrated in [35]. In details, we learn the feature map $\mathbf{f}$ and the TQM $\mathbf{Q}$ *simultaneously* by minimizing the following objective:

$$\min_{\mathbf{f},\mathbf{Q}} \ \gamma \mathsf{KL}(\mathbf{f}_{\#}p \| \mathbf{Q}_{\#}q) + \lambda \ell(\mathbf{f}) + \zeta g(\mathbf{Q}), \tag{2.3}$$

where g embodies some potential constraints on the increasing triangular map $\mathbf{Q}$, ℓ is some loss associated with learning the feature map $\mathbf{f}$, q is a fixed reference density (in our case the uniform density over the hypercube $[0,1]^m$), $\zeta, \lambda, \gamma \geq 0$ are regularization constants, and we use the KL-divergence to measure the discrepancy between two densities.

Exploiting Proposition 1 we parameterize the TQM as the composition $\mathbf{Q} = \mathbf{T} \circ \mathbf{\Phi}^{-1}$, where $\mathbf{\Phi} = (\Phi, \ldots, \Phi)$ with Φ the CDF of standard univaraite Gaussian and $\mathbf{T} : \mathbb{R}^d \to \mathbb{R}^d$ an increasing triangular map. Note that unlike $\mathbf{Q}$ whose support is constrained to the unit hypercube, there is no constraint on the support of $\mathbf{T}$, hence it is easier to handle the latter computationally.

2.3 Novelty Detection Rules

Once the feature map $\mathbf{f}$ and TQM $\mathbf{Q}$ are estimated (see §3), we can detect novel test samples by either thresholding the density function of the latent variable $\mathbf{Z}$ or thresholding its TQM.

[1] The notation $\mathbf{T}_{\#}p$ stands for the push-forward density, i.e., the density of $\mathbf{T}(\mathbf{X})$ when $\mathbf{X} \sim p$.

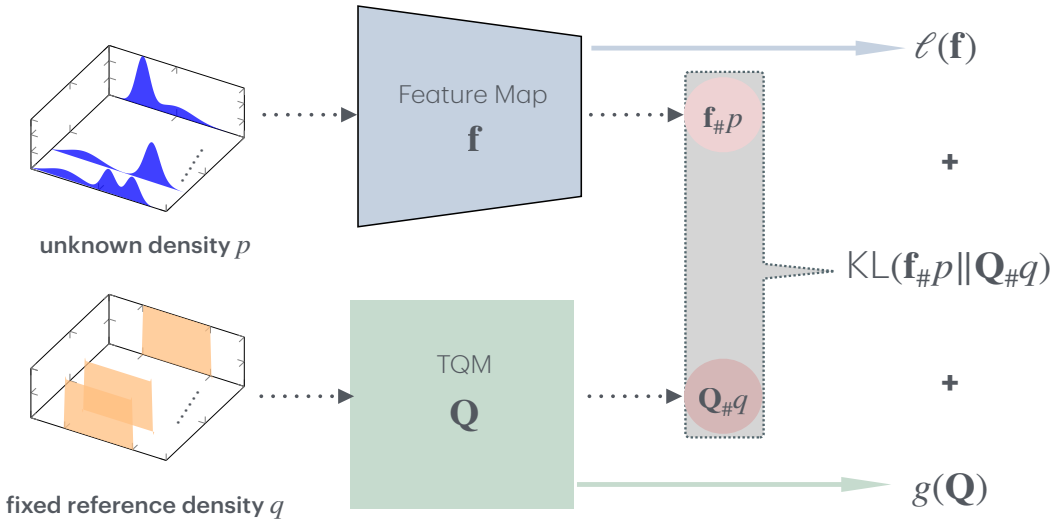

Figure 2.3: Schematic of our general framework for novelty detection.

2.3.1 Density Scoring Rule

The density of $\mathbf{Z} = \mathbf{f}(\mathbf{X}) = \mathbf{Q}(\mathbf{U}) = \mathbf{T}(\mathbf{\Phi}^{-1}(\mathbf{U}))$, using the change-of-variable formula, is

$$p_{\mathbf{Z}}(\mathbf{z}) = 1/|\mathbf{Q}'(\mathbf{Q}^{-1}(\mathbf{z}))| = \tfrac{1}{|\mathbf{T}'(\mathbf{T}^{-1}(\mathbf{z}))|} \cdot \prod_{j=1}^{m} \varphi([\mathbf{T}^{-1}(\mathbf{z})]_j), \quad \text{where} \quad \varphi = \Phi'. \tag{2.4}$$

Thus, we declare a test sample $\tilde{\mathbf{X}}$ to be "novel" if

$$\log|\mathbf{T}'(\mathbf{T}^{-1}(\mathbf{f}(\tilde{\mathbf{X}})))| + \tfrac{1}{2}\|\mathbf{T}^{-1}(\mathbf{f}(\tilde{\mathbf{X}}))\|_2^2 \geq \tau, \tag{2.5}$$

where τ is some chosen threshold. Crucially, since $\mathbf{T}$ is increasing triangular, $\mathbf{T}^{-1}$ and the triangular determinant $|\mathbf{T}'|$ can both be computed very efficiently [35]. The (slight) downside of this density approach is that the scale of an appropriate threshold τ is usually difficult to guess.

2.3.2 Quantile Scoring Rule

Alternatively, we can declare a test sample $\tilde{\mathbf{X}}$ to be "novel" by directly thresholding the TQM $\mathbf{Q}$. Indeed, let $N \subseteq [0,1]^m$ be a subset whose (uniform) measure is $1-\alpha$ for some $\alpha \in (0,1)$, then we say $\tilde{\mathbf{X}}$ is "novel" iff

$$\mathbf{Q}^{-1}(\mathbf{f}(\tilde{\mathbf{X}})) \notin N. \tag{2.6}$$

For instance, we can choose N to be the cube centered at $(1/2, \ldots, 1/2)$ and with side length $(1-\alpha)^{1/m}$, in which case

$$\mathbf{Q}^{-1}(\mathbf{f}(\tilde{\mathbf{X}})) \notin N \iff \|\mathbf{Q}^{-1}(\mathbf{f}(\tilde{\mathbf{X}})) - \tfrac{1}{2}\|_\infty \geq (1-\alpha)^{1/m}/2. \tag{2.7}$$

The upside of this quantile approach is that we can control Type-I error (i.e. false positive) precisely, i.e. if $\tilde{\mathbf{X}}$ is indeed sampled from p, then we will declare it to be novel with probability at most α.

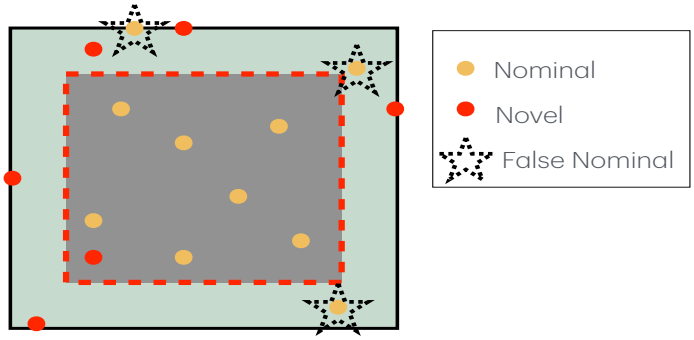

Figure 2.4: Illustration of controlled Type-I error by using the quantile scoring rule.

In Figure 2.4, we present a simplified example using a quantile with two dimensions to illustrate the application of the quantile scoring rule. After we get the estimated $\hat{\mathbf{Q}}$, we just thresholding the 2D quantile directly. As showed in the figure, we designate the inner square (e.g., area = 0.9) as the selected region N. If the quantile of the test point $\hat{\mathbf{Q}}^{-1}(f(\tilde{\mathbf{X}}_t))$ falls within this square, we classify $\mathbf{X}_t$ as "nominal"; otherwise, it is classified as "novel". In an ideal scenario, where $\hat{\mathbf{Q}}$ is estimated accurately, the probability that the quantile of a "nominal" sample falls within $[0,1]^2$ is uniform across all regions. Consequently, for a "nominal" test sample, the probability of its quantile falling outside the selected square is less than 0.1, precisely representing the controlled false positive rate (i.e., we will declare a nominal test point to be novel with probability at most 0.1). This method not only outputs a prediction but also controls the risk of the prediction.

The quantile scoring rule prompts us to explore a broader and more intriguing question: How can we precisely quantify the accuracy and uncertainty of predictions made by complex models, such as deep neural networks? Typically, this question receives only a rough approximation. For instance, in this thesis, the control of Type-I error relies on the assumption that we have accurately learned the TQM $\mathbf{Q}$, a condition that is challenging to ensure in practice. Consequently, the

control over Type-I error is only approximate. To address this challenge, numerous works have been proposed to solve this question, especially in the field of conformal inference [53], [54] provides a general distribution-free method to rigorously calibrate the output of any machine learning algorithm for novelty detection.

2.4 Advantages of Our General Framework

Before proceeding to the implementation details of our framework (2.3), let us mention the advantages of our general framework (2.3) for novelty detection:

(a) It allows us to perform feature extraction on the original sample $\mathbf{X}$ in an end-to-end fashion. As is well-known, density estimation hence also novelty detection becomes extremely challenging when the dimension d is high. Our framework alleviates this curse-of-dimensionality by setting $m \ll d$ and employing $\mathbf{f}$ to perform dimensionality reduction.

(b) Our end-to-end framework enables us to adopt the recent flow-based density estimation algorithms, which have been shown to be universally consistent [55, 35] and extremely effective in practice.

(c) By estimating the TQM $\mathbf{Q}$ once, we can employ the two scoring rules, i.e. the density scoring rule (2.5) and the quantile scoring rule (2.6), simultaneously, without incurring any extra overhead. This allows us to perform a fair and comprehensive experimental comparison of the two complementary approaches.

(d) Last but not least, our framework recovers, unifies, and extends many existing approaches in the literature.

In this section, we will introduce three classic approaches that can be unified under our general framework.

Example 1 (One-class SVM [17]) *As shown in [56], the one-class SVM minimizes precisely the conditional value-at-risk, which is the average of the tail of a distribution:*

$$\min_{f} \ \mathtt{CVaR}_{\alpha}(f(\mathbf{X})) + \lambda \|f\|_{\mathcal{H}_\kappa}^2, \quad \textit{where} \quad \mathtt{CVaR}_{\alpha}(Z) := \mathsf{E}(Z | Z \geq Q_Z(\alpha)), \tag{2.8}$$

$Q_Z(\alpha)$ is the α-th quantile of the real random variable Z, and $\mathcal{H}_\kappa$ is the reproducing kernel Hilbert space (RKHS) induced by some kernel κ. This approach employs the quantile scoring rule (2.6).

To cast one-class SVM into our framework (2.3), let us set $m = 1$ hence the TQM reduces to the classical one. Let $\ell(f) = \|f\|^2_{\mathcal{H}_\kappa}$ and $g(\mathbf{Q}) = \texttt{CVaR}_\alpha(\mathbf{Q}_{\#}q)$. Now with $\zeta = 1$ and $\gamma = \infty$ in (2.3) we recover the celebrated one-class SVM.

If instead of choosing f from an RKHS, we represent f using a deep network, then we recover the recent approach in [57].

Example 2 (Support Vector Data Description (SVDD) [18]) *Similar to one-class SVM, it is easy to show that SVDD also minimizes the conditional value-at-risk:*

$$\min_{\mathbf{c} \in \mathcal{H}_\kappa} \ \texttt{CVaR}_\alpha(\|\boldsymbol{\varphi}(\mathbf{X}) - \mathbf{c}\|^2_{\mathcal{H}_\kappa}), \tag{2.9}$$

where $\boldsymbol{\varphi} : \mathbb{R}^d \to \mathcal{H}_\kappa$ is the canonical feature map of the RKHS. This approach also employs the quantile scoring rule (2.6). It is well-known known that SVDD and one-class SVM are equivalent for radial kernels [e.g. 17].

Again in this case $m = 1$. Let $f(\mathbf{X}) = \|\boldsymbol{\varphi}(\mathbf{X}) - \mathbf{c}\|^2_{\mathcal{H}_\kappa}$, $\ell \equiv 0$ and $g(\mathbf{Q}) = \texttt{CVaR}_\alpha(\mathbf{Q}_{\#}q)$. As γ approaches ∞ in (2.3), we recover the SVDD formulation.

If instead of choosing $\boldsymbol{\varphi}$ as the canonical feature map of an RKHS, we represent $\boldsymbol{\varphi}$ using a deep network, then we recover the recent approach in [25].

Example 3 (Latent Space Autoregression (LSA) [31]) *The recent work [31], following a sequence of previous attempts [24, 26, 22, 21], proposed to learn the feature map $\mathbf{f}$ using an auto-encoder structure, and to learn the density of the latent variable $\mathbf{Z} = \mathbf{f}(\mathbf{X})$ using an autoregressive model, which, as argued in [35], exactly corresponds to a triangular map. In other words, if we set $\mathbf{f}$ as the parameters of an auto-encoder, ℓ to be its reconstruction loss, and $g \equiv 0$, then our framework (2.3) reduces to LSA. However, our general framework opens the way to exploit more advanced flow-based density estimation algorithms, as well as the quantile scoring rule (2.6).*

Chapter 3

Estimating TQM Using Deep Networks

In this chapter, we will present the details of implementing our general framework (2.3) for novelty detection. In §3.1, we give a specific case of implementing our general framework, which consists of three components: a feature extractor, a flow-based neural density estimator, and the empirically estimated KL-divergence term. Specially, we introduce how to estimate the TQM $\mathbf{Q}$ in (2.3) based on samples $\{\mathbf{X}_1, \ldots, \mathbf{X}_n\} \overset{i.i.d.}{\sim} p$. In particular, any flow-based neural density estimator can be plugged into our framework. In §3.2, we cast the two competing objectives in (2.3) as multi-objective optimization and apply the Multiple Gradient Descent Algorithm (MGDA) to automatically tune the trade-off hyperparameter in each iteration.

3.1 Details of Our Implementations

Our framework (2.3) not only recovers, but also unifies and extends numerous existing approaches documented in the literature. Consequently, there exist various implementation of each component within our framework, depended on the deployed models and applications. In this section, we present our implementation (see Figure 3.1), which other deep feature extractors and flow-based neural density estimator can be plugged into.

(1) **A feature extractor $\mathbf{f}$ for performing dimensionality reduction.** Following previous works [31, 24, 26, 22, 21] we implement $\mathbf{f}$ through a deep autoencoder that consists of one encoder $\mathbf{Z} = \mathcal{E}(\mathbf{X}; \boldsymbol{\theta}_E)$ and one decoder $\hat{\mathbf{X}} = \mathcal{D}(\mathbf{Z}; \boldsymbol{\theta}_D)$. We use the Euclidean reconstruction loss:

$$\ell(\mathbf{f}) = \ell(\boldsymbol{\theta}_E, \boldsymbol{\theta}_D) = \textstyle\sum_{i=1}^{n} \|\mathbf{X}_i - \hat{\mathbf{X}}_i\|^2. \tag{3.1}$$

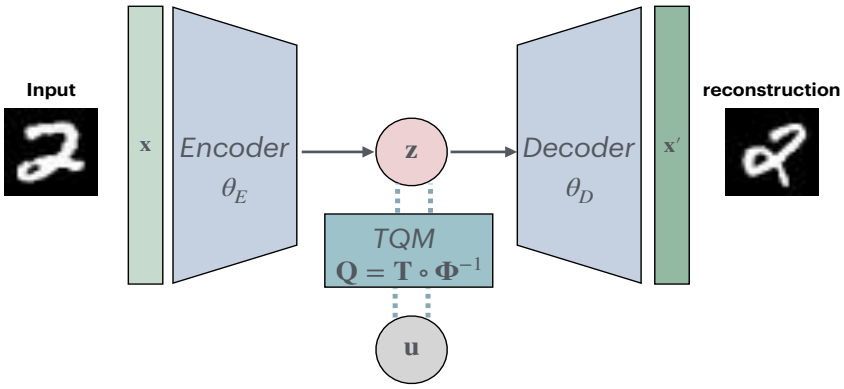

Figure 3.1: Implementation of our general framework for novelty detection.

As argued in [58], the reconstruction error, aside from low likelihood, is an important indicator for "novelty." Indeed, since the autoencoder is trained on nominal data, a test sample will incur a large reconstruction error only when it is novel, as such samples have never been encountered before.

(2) **A flow-based neural density estimator for $\mathbf{Q}$.** Here we adopt the sum-of-squares (SOS) flow proposed in [35], although other neural density estimators would apply equally well. The SOS flow consists of two parts: an increasing (univariate) polynomial $\mathfrak{P}_{2r+1}(u; \mathbf{a})$ with degree $2r+1$ for modelling conditional densities and a conditioner network $C_j(u_1, \ldots, u_{j-1}; \boldsymbol{\theta}_Q)$ for generating the coefficients $\mathbf{a}$ of the polynomial:

$$\mathfrak{P}_{2r+1}(u; \mathbf{a}) = c + \int_0^u \sum_{s=1}^k \left(\sum_{l=0}^r a_{l,s} t^l \right)^2 \mathrm{d}t, \tag{3.2}$$

where $c \in \mathbb{R}$ is an arbitrary constant, $r \in \mathbb{N}$ is the degree of polynomial, and k can be chosen as small as 2. In other words, the TQM $\mathbf{Q}$ learned using SOS flow has the following form:

$$\mathbf{Q} = \mathbf{T} \circ \mathbf{\Phi}^{-1}, \quad \text{where} \quad \forall j, \;\; T_j(u_1, \ldots, u_j) = \mathfrak{P}_{2r+1}\big(u_j; C_j(u_1, \ldots, u_{j-1}; \boldsymbol{\theta}_Q)\big). \tag{3.3}$$

Any regularization term on the conditioner network weights $\boldsymbol{\theta}_Q$ can be put into the function $g(\mathbf{Q})$ in our framework (2.3).

(3) Lastly, **the KL-divergence term** in (2.3) can be approximated empirically using the given sample $\lbag \mathbf{X}_1, \ldots, \mathbf{X}_n \rbag$. Upon dropping irrelevant constants we reduce the KL term in (2.3) to:

$$\min_{\boldsymbol{\theta}_Q} \;\; \sum_{i=1}^n \Big[\log |\mathbf{Q}'(\mathbf{Q}^{-1}(\mathbf{f}(\mathbf{X}_i)))| - \log q(\mathbf{Q}^{-1}(\mathbf{f}(\mathbf{X}_i))) \Big], \tag{3.4}$$

where each component of $\mathbf{Q}$ is given in (3.3). Crucially, since $\mathbf{Q}$ is increasing triangular, evaluating the inverse $\mathbf{Q}^{-1}$ and the Jacobian $|\mathbf{Q}'|$ can both be done in linear time [35].

Since q is the uniform density over the hypercube, upon simplification the final training objective we use in our experiments is as follows. Let $\mathbf{Z}_i = \mathcal{E}(\mathbf{X}_i; \boldsymbol{\theta}_E)$, we aim to solve:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} (1-\lambda) \Big[\underbrace{\log |\mathbf{T}'(\mathbf{T}^{-1}(\mathbf{Z}_i))| + \|\mathbf{T}^{-1}(\mathbf{Z}_i)\|_2^2/2}_{\text{negative log-likelihood } h(\mathbf{X}_i;\boldsymbol{\theta})} \Big] + \lambda \underbrace{\|\mathbf{X}_i - \mathcal{D}(\mathbf{Z}_i; \boldsymbol{\theta}_D)\|^2}_{\text{reconstruction loss } \ell(\mathbf{X}_i;\boldsymbol{\theta})}, \tag{3.5}$$

and recall that $\mathbf{Q} = \mathbf{T} \circ \boldsymbol{\Phi}^{-1}$ is parameterized through the conditioner network weights $\boldsymbol{\theta}_Q$ in (3.3). We did not find it necessary to further regularize $\mathbf{Q}$ hence set $g \equiv 0$ in (2.3) and w.l.o.g. $\gamma = 1 - \lambda$.

3.2 MGDA for auto-tuning hyperparameters

The first KL term in (2.3), as is well-known, reduces to the negative log-likelihood of the latent random vectors $\mathbf{Z}_i$ in (3.5), and the second term is the standard reconstruction loss. The two terms share the encoder weights $\boldsymbol{\theta}_E$ and the trade-off is balanced through the hyperparameter λ. This design choice conforms to the psychology findings in [58]. In practice, we found that the variance of the log-likelihood is much larger than that of the reconstruction loss, and as a consequence we observed substantial difficulty in directly minimizing the weighted objective in (3.5).

A popular pre-training heuristic is to train the whole model in two stages: we first minimize the reconstruction loss $\ell(\boldsymbol{\theta}_E, \boldsymbol{\theta}_D)$ and then, with the learned hidden vector $\mathbf{Z}$, we estimate the TQM $\mathbf{Q}$ by maximum likelihood. However, as shown in [22], the latent representation learned in the first stage does not necessarily help the task in the second stage.

Instead, we cast the two competing objectives in (3.5) as multi-objective optimization, which we solve using the multiple gradient descent algorithm (MGDA) [38, 39, 40]. Our motivation comes from the following observation: the two-stage procedure amounts to first setting $\lambda = 1$ and running gradient descent (GD) for a number of iterations, then switching to $\lambda = 0$ (or $\lambda = 0.5$ say) and running GD for the remaining iterations.

Naturally, instead of any pre-determined schedule for the hyperparameter λ (such as switching from 1 to 0 or 0.5), why not let GD decide what λ to use in each iteration? This is precisely the main idea behind MGDA, where at iteration t we solve

$$\lambda_t = \operatorname*{argmin}_{0 \le \lambda \le 1} \left\| \sum_{i \in I} (1-\lambda) \nabla h(\mathbf{X}_i; \boldsymbol{\theta}_t) + \lambda \nabla \ell(\mathbf{X}_i; \boldsymbol{\theta}_t) \right\|^2 = \min \left\{ 1, \max \left\{ 0, \frac{\langle \nabla h_I - \nabla \ell_I, \nabla h_I \rangle}{\|\nabla h_I - \nabla \ell_I\|^2} \right\} \right\}, \tag{3.6}$$

where $I \subseteq \{1, \ldots, n\}$ is a minibatch of samples, and obviously $\nabla h_I = \sum_{i \in I} \nabla h(\mathbf{X}_i; \boldsymbol{\theta}_t)$ and similarly for $\nabla \ell_I$. With λ_t calculated we can continue the gradient update:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta[(1 - \lambda_t)\nabla h_I + \lambda_t \nabla \ell_I], \tag{3.7}$$

where $\eta \geq 0$ is the step size. As shown in [40], this algorithm converges to a Pareto-optimal solution under fairly general conditions. Pleasantly, MGDA eliminates the need of tuning the hyperparameter λ as it is determined automatically on the fly. To our best knowledge, our work is the first to demonstrate the effectiveness of MGDA on novelty detection tasks.

We end our discussion by pointing out that the algorithm we develop here can easily be adapted to other design choices that fit into our general framework (2.3). For instance, if we use a variational autoencoder [59] or a denoising autoencoder [60], then we need only replace the square reconstruction loss in (3.5) accordingly.

Chapter 4

Evaluation

In this chapter, we present experiments to demonstrate the effectiveness of our proposed framework. In §4.1, we first introduce the experimental setting up, including the variants of our framework, datasets, baselines and the related performance metrics. In §4.2, we then compare five variants of our proposed method with different scoring functions, to show that our framework's general effectiveness for common scores used in novelty detection. In §4.3, We conduct experiments testing various training strategies, and the results demonstrate that our MGDA strategy outperforms other two-stage training strategies. Lastly, in §4.4, we evaluate the effectiveness of our proposed novelty detection method by comparing it with traditional and state-of-the-art alternatives. Additionally, we include visualization results in §4.5 to enhance our understanding of the performance of our methods.

4.1 Experimental Settings

4.1.1 Variants of Our Method

In this subsection, we first introduce some variants of our proposed method based on different scoring function. We consider the following five scoring functions that we threshold at some level τ. In particular, given a test example $\tilde{\mathbf{X}}$, we denote its reconstruction by $\hat{\mathbf{X}}$ and its latent representation by $\tilde{\mathbf{Z}} = \mathbf{f}(\tilde{\mathbf{X}})$.

(1) Reconstruction error (`REC`): $\|\tilde{\mathbf{X}} - \hat{\mathbf{X}}\|^2$;

(2) Negative log-likelihood (`NLL`): $\log|\mathbf{T}'(\mathbf{T}^{-1}(\tilde{\mathbf{Z}}))| + \|\mathbf{T}^{-1}(\tilde{\mathbf{Z}})\|_2^2/2$;

(3) 1-norm of quantile ($\texttt{TQM}_1$): $\|\mathbf{\Phi}(\mathbf{T}^{-1}(\tilde{\mathbf{Z}})) - \frac{1}{2}\|_1$,

(4) 2-norm of quantile ($\texttt{TQM}_2$): $\|\mathbf{\Phi}(\mathbf{T}^{-1}(\tilde{\mathbf{Z}})) - \frac{1}{2}\|_2$;

(5) Infinity norm of quantile ($\texttt{TQM}_\infty$): $\|\mathbf{\Phi}(\mathbf{T}^{-1}(\tilde{\mathbf{Z}})) - \frac{1}{2}\|_\infty$.

4.1.2 Datasets and Competitor Algorithms

In our experiments, we use two public image datasets: MNIST and Fashion-MNIST, as well as two non-image datasets: KDDCUP and Thyroid. A detailed description of these datasets, the applied network architectures, and the training hyperparameters can be found in Appendix A. For MNIST and Fashion-MNIST, each of the ten classes is deemed as the nominal class while the rest of the nine classes are deemed as the novel class. We use the standard training and test splits. For every class, we hold out 10% of the training set as the validation set, which is used to tune hyperparameters and to monitor the training process.

We compare our method with the following alternative algorithms:

- **OC-SVM** [17]. OC-SVM is a traditional kernel-based quantile approach which has been widely used in practice for novelty detection. We use the RBF kernel in our experiments. We consider two OC-SVM-based methods for comparison. 1) RAW-OC-SVM: the input is directly fed to OC-SVM; 2) CAE-OC-SVM: a convolutional autoencoder is first applied to the input data for dimensionality reduction, and then the low-dimensional latent representation is fed to OC-SVM.

- **Geometric transformation (GT)** [28]. A self-labeled multi-class dataset is first created by applying a set of geometric transformations to the original nominal examples. Then, a multi-class classifier is trained to discriminate the geometric transformations of each nominal example. The scoring function in GT is the conditional probability of the softmax responses of the classifier given the geometric transformations.

- **Variational autoencoder (VAE)** [59]. The evidence lower bound is used as the scoring function.

- **Denoising autoencoder (DAE)** [60]. The reconstruction error is used as the scoring function.

- **Deep structured energy-based models (DSEBM)** [21]. DSEBM employs a deterministic deep neural network to output the energy function (i.e., negative log-likelihood), which is used to form the density of nominal data. The network is trained by score matching in

a way similar to training DAE. Two scoring functions based on reconstruction error and energy score are considered.

- **Deep autoencoding Gaussian mixture model (DAGMM)** [22]. DAGMM consists of a compression network implemented using a deep autoencoder and a Gaussian mixture estimation network that outputs the joint density of the latent representations and some reconstruction features from the autoencoder. The energy function is used as the scoring function.
- **Generative probabilistic novelty detection (GPND)** [26]. GPND, based on adversarial autoencoders, employs an extra adversarial loss to impose priors on the output distribution. The density is used as the scoring function. By linearizing the manifold that nominal data resides on, its density is factorized into two product terms, which are then approximately computed using nominal data.
- **Latent space autoregression (LSA)** [31]. A parametric autoregressive model is used to estimate the density of the latent representation generated by a deep autoencoder, where the conditional probability densities are modeled as multinomials over quantized latent representations. The sum of the normalized reconstruction error and log-likelihood is used as the scoring function.

4.1.3 Performance Metrics

For evaluation, we use precision, recall, F1 score, and the Area Under Receiver Operating Characteristic (AUROC) curve as our performance metrics, which are commonly used in previous works. Same as the setting of novelty detection, we take "novelty" as positive cases and "nominal" cases as negative cases.

4.2 Comparing Variants of Our Method

In Table 4.1, we compare two approaches on MNIST for selecting the hyperparameter λ in the training phase: 1) chosen from a pre-set family using the validation set; and 2) automatically optimized using MGDA [38, 39, 40]. We report the average AUROC over 10 classes. It is clear that for all scoring functions, the optimized λ generally leads to the highest AUROC. This is also observed on other datasets such as Fashion-MNIST. Within the proposed variants, NLL results in the highest AUROC among all scoring functions, followed by TQM_2.

In Table 4.2, on the two non-image datasets we evaluate the average precision, recall, and F1 score. The superscript $*$ on the baselines indicates that the results are directly quoted from the respective references. The threshold is chosen by assuming the prior knowledge of the ratio between the novel and nominal examples in the test set. Under this assumption, the number of false positives is equal to that of false negatives, thus the value of the three metrics coincides. On Thyroid, $\texttt{TQM}_\infty$ is slightly better than the density-based method. On KDDCUP, the density and quantile-based approaches have the same performance, while REC results in the worst performance. On both datasets, our proposed methods are superior to the benchmarks.

Table 4.1: AUROC of Variants of Our Method on MNIST

Scoring function	$\lambda = 0.99$	0.9	0.5	0.1	Optimized
NLL	**0.9729**	0.9692	0.9537	0.9389	0.9728
$\texttt{TQM}_1$	0.9622	0.9616	0.9430	0.9319	**0.9666**
$\texttt{TQM}_2$	0.9666	0.9645	0.9465	0.9347	**0.9699**
$\texttt{TQM}_\infty$	0.9499	0.9527	0.9371	0.9128	**0.9531**

Table 4.2: Average Precision, Recall, and F1 Score on Non-image Datasets

	Thyroid			KDDCUP		
Method	Precision	Recall	F1	Precision	Recall	F1
RAW-OC-SVM *	0.3639	0.4239	0.3887	0.7457	0.8523	0.7954
DSEBM *	0.0404	0.0403	0.0403	0.7369	0.7477	0.7423
DAGMM *	0.4766	0.4834	0.4782	0.9297	0.9442	0.9369
Ours-REC	–	–	–	0.6305	0.6287	0.6296
Ours-NLL	0.7312	0.7312	0.7312	**0.9622**	**0.9622**	**0.9622**
Ours-$\texttt{TQM}_1$	0.5269	0.5269	0.5269	0.9621	0.9621	0.9621
Ours-$\texttt{TQM}_2$	0.5806	0.5806	0.5806	**0.9622**	**0.9622**	**0.9622**
Ours-$\texttt{TQM}_\infty$	**0.7527**	**0.7527**	**0.7527**	**0.9622**	**0.9622**	**0.9622**

4.3 Comparing MGDA with two-stage training strategies

In our proposed algorithm the autoencoder and the estimation network are trained jointly by employing MGDA. For comparison, we also consider the following two-stage training strategies:

(1) We first train the autoencoder, then fix the autoencoder and train the estimation network alone (denoted as **Fix-**).

(2) We first pretrain the autoencoder, then jointly train the autoencoder and the estimation network with the weight λ fixed to 0.5 (denoted as **Pretrain-**).

Table 4.3: Comparison between joint and two-stage training: AUROC on MNIST

Class	Fix-NLL	Pretrain-NLL	Ours-NLL	Fix-TQM_2	Pretrain-TQM_2	Ours-TQM_2
0	0.9939	**0.9954**	0.9951	0.9904	**0.9939**	0.9925
1	0.9971	**0.9988**	0.9977	0.9972	**0.9985**	0.9969
2	0.9403	**0.9677**	0.9526	0.9188	**0.9568**	0.9479
3	0.9568	0.9496	**0.9627**	**0.9481**	0.9414	0.9567
4	**0.9703**	0.9445	0.9657	**0.9700**	0.9388	0.9625
5	0.9612	0.9564	**0.9618**	0.9525	0.9486	**0.9601**
6	0.9878	0.9907	**0.9915**	0.9841	0.9881	**0.9895**
7	0.9629	0.9676	**0.9686**	0.9587	0.9656	**0.9660**
8	0.9549	**0.9587**	0.9551	0.9397	**0.9527**	0.9512
9	0.9736	0.9733	**0.9768**	0.9742	0.9641	**0.9756**
avg	0.9699	0.9703	**0.9728**	0.9634	0.9649	**0.9699**

The comparison regarding AUROC on MNIST is shown in Table 4.3. We found that the proposed joint training method leads to the best performance for both the density-based and the quantile-based scoring functions. This is consistent with the findings in many existing works [e.g. 31, 57, 25, 21]. For the fixed two-stage method, our understanding is that the latent representation learned in the first stage may not be the most beneficial for the training of the estimation network in the second stage, which in turn degrades the overall performance. For the pretrained two-stage method, although in the second stage the two parts are trained jointly the autoencoder is initialized with the parameters learned in the first stage, which might prevent it from being updated to a more suitable local optimum.

In Table 4.4, we show the comparison between joint and two-stage training on Fashion-MNIST dataset. The observation is similar to that on MNIST dataset.

Table 4.4: Comparison between joint and two-stage training: AUROC on Fashion-MNIST

Class	Fix-NLL	Pretrain-NLL	Ours-NLL	Fix-TQM_2	Pretrain-TQM_2	Ours-TQM_2
0	0.9114	0.8612	**0.9217**	0.8959	0.8650	**0.9169**
1	0.9764	**0.9852**	0.9579	0.9639	**0.9813**	0.9496
2	0.8799	0.8575	**0.8985**	0.8809	0.8548	**0.8990**
3	**0.9370**	0.9222	0.9304	0.9269	0.9233	**0.9245**
4	0.9013	0.9132	**0.9223**	0.8859	0.9080	**0.9209**
5	0.9096	**0.9117**	0.8940	**0.9140**	0.9098	0.8844
6	0.8424	0.7488	**0.8435**	**0.8391**	0.7617	0.8384
7	0.9757	**0.9842**	0.9802	0.9689	**0.9843**	0.9718
8	0.9125	0.8851	**0.9450**	0.8962	0.8750	**0.9429**
9	0.9776	**0.9879**	0.9825	0.9780	0.9827	**0.9830**
avg	0.9224	0.9057	**0.9276**	0.9150	0.9046	**0.9234**

4.4 Comparing with Baselines

In this section, we compare our method with the baseline approaches. Note that except RAW-OC−SVM and GT, all other methods, including our own variants, are based on autoencoders.

In Table 4.5, we show the comparison of AUROC on the image datasets. Among the proposed quantile scoring functions we only list TQM_2, which outputs the highest value of AUROC. We observe that on both datasets our proposed methods are superior to most of the benchmarks, with the density scoring function being slightly better than the quantile one. On MNIST, GPND and GT have better performance; and on Fashion-MNIST, GT outputs the highest value of AUROC followed by Ours-NLL and RAW-OC−SVM. However, since GT explicitly extracts features by using a set of geometric transformations, it inevitably suffers a high computational and space complexity.

4.5 Visualization

In this section, we show visualization results on the MNIST dataset. We use digit 1 as the nominal class. The results for other classes are similar. These visualizations can be used for diagnosing the training process and for assessing the quality of the learned TQM: by definition, the pre-image of data under TQM should be uniformly distributed on the hypercube $[0, 1]^m$.

In Figure 4.3, we show the violin plots of the scoring statistics NLL, TQM_1, TQM_2, and TQM_∞

Table 4.5: AUROC on MNIST and Fashion-MNIST

Class	OC-SVM RAW	OC-SVM CAE	VAE	DAE	LSA	GT	DAGMM	GPND	DSEBM	Ours-NLL	Ours-$\mathtt{TQM}_2$
	MNIST										
0	0.995	0.990	0.985	0.982	0.998	0.982	0.500	0.999	0.320	0.995	0.993
1	0.999	0.999	0.997	0.998	0.999	0.893	0.766	0.999	0.987	0.998	0.997
2	0.926	0.919	0.943	0.936	0.923	0.993	0.326	0.980	0.482	0.953	0.948
3	0.936	0.939	0.916	0.929	0.974	0.987	0.319	0.968	0.753	0.963	0.957
4	0.967	0.946	0.945	0.940	0.955	0.993	0.368	0.980	0.696	0.966	0.963
5	0.955	0.936	0.929	0.928	0.966	0.994	0.490	0.987	0.727	0.962	0.960
6	0.987	0.979	0.977	0.982	0.992	0.999	0.515	0.998	0.954	0.992	0.990
7	0.966	0.951	0.975	0.971	0.969	0.966	0.500	0.988	0.911	0.969	0.966
8	0.903	0.896	0.864	0.857	0.935	0.974	0.467	0.929	0.536	0.955	0.951
9	0.962	0.960	0.967	0.974	0.969	0.993	0.813	0.993	0.905	0.977	0.976
avg	0.960	0.952	0.950	0.950	0.968	0.977	0.508	**0.982**	0.727	0.973	0.970
	Fashion-MNIST										
0	0.919	0.908	0.874	0.867	0.916	0.903	0.303	0.917	0.891	0.922	0.917
1	0.990	0.987	0.977	0.978	0.983	0.993	0.311	0.983	0.560	0.958	0.950
2	0.894	0.884	0.816	0.808	0.878	0.927	0.475	0.878	0.861	0.899	0.899
3	0.942	0.911	0.912	0.914	0.923	0.906	0.481	0.945	0.903	0.930	0.925
4	0.907	0.913	0.872	0.865	0.897	0.907	0.499	0.906	0.884	0.922	0.921
5	0.918	0.865	0.916	0.921	0.907	0.954	0.413	0.924	0.859	0.894	0.884
6	0.834	0.820	0.738	0.738	0.841	0.832	0.420	0.785	0.782	0.844	0.838
7	0.988	0.984	0.976	0.977	0.977	0.981	0.374	0.984	0.981	0.980	0.972
8	0.903	0.877	0.795	0.782	0.910	0.976	0.518	0.916	0.865	0.945	0.943
9	0.982	0.955	0.965	0.963	0.984	0.994	0.378	0.876	0.967	0.983	0.983
avg	0.928	0.910	0.884	0.881	0.922	**0.937**	0.472	0.911	0.855	0.928	0.923

on MNIST test set (with digit 1 serving the nominal class). We use the network parameters produced at every 20 epochs in training to generate each curve. We can see that, in the beginning the nominal and novel data have a large region of overlap and after more training epochs they are gradually separated. After about 20 epochs of training they can be clearly distinguished under $\mathtt{NLL}$, $\mathtt{TQM}_1$, and $\mathtt{TQM}_2$, which indicates the effectiveness of these scoring functions. For $\mathtt{TQM}_\infty$, the distribution of novel data is concentrated within a narrow region, which is near the boundary of that of nominal data.

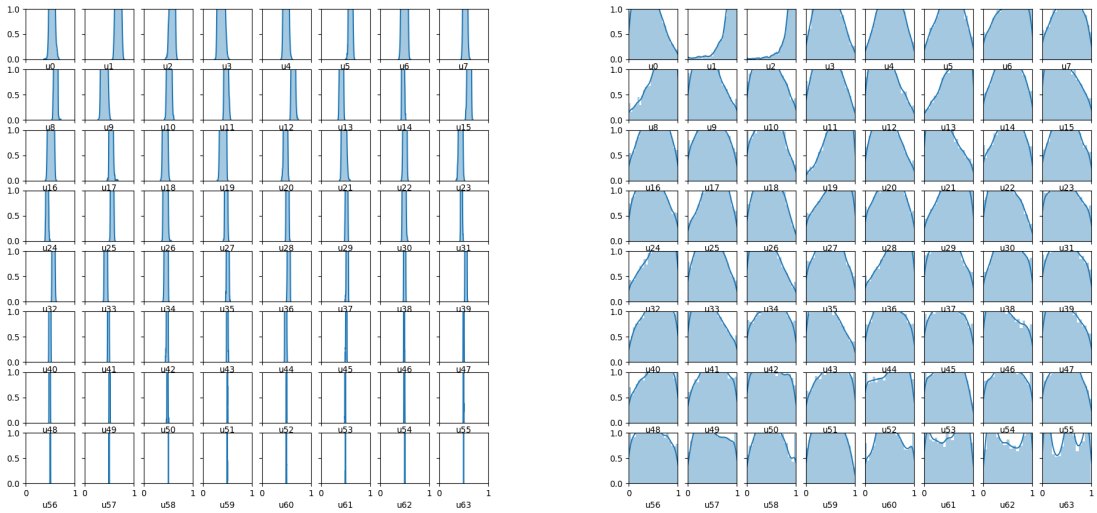

Figure 4.1: All marginals of pre-image of training data in $[0,1]^{64}$: 1) marginals at initialization; and 2) marginals at 1000 epochs of training.

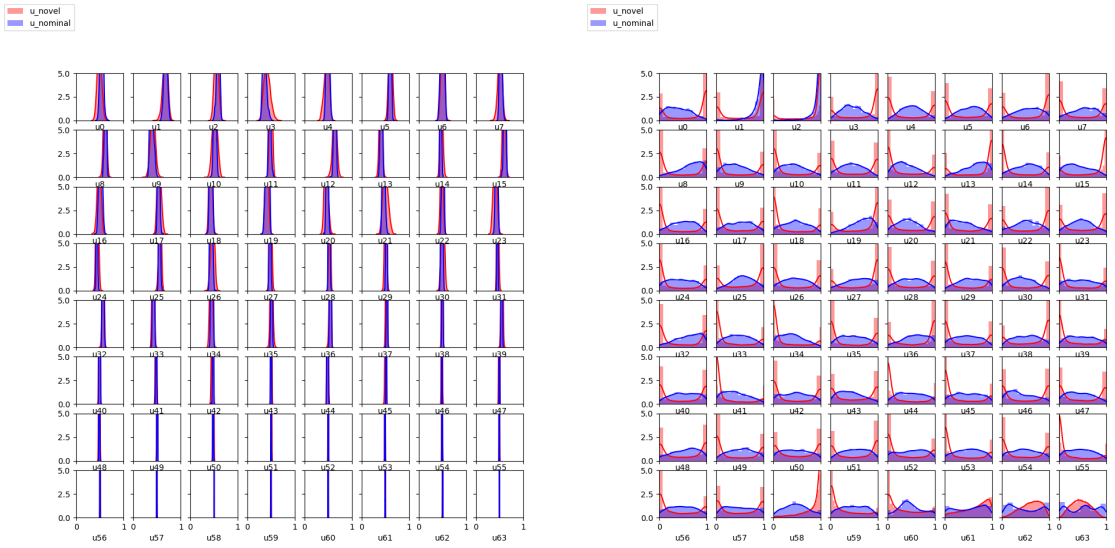

Figure 4.2: All marginals of pre-image of test data in $[0,1]^{64}$: 1) marginals at initialization; and 2) marginals at 1000 epochs of training.

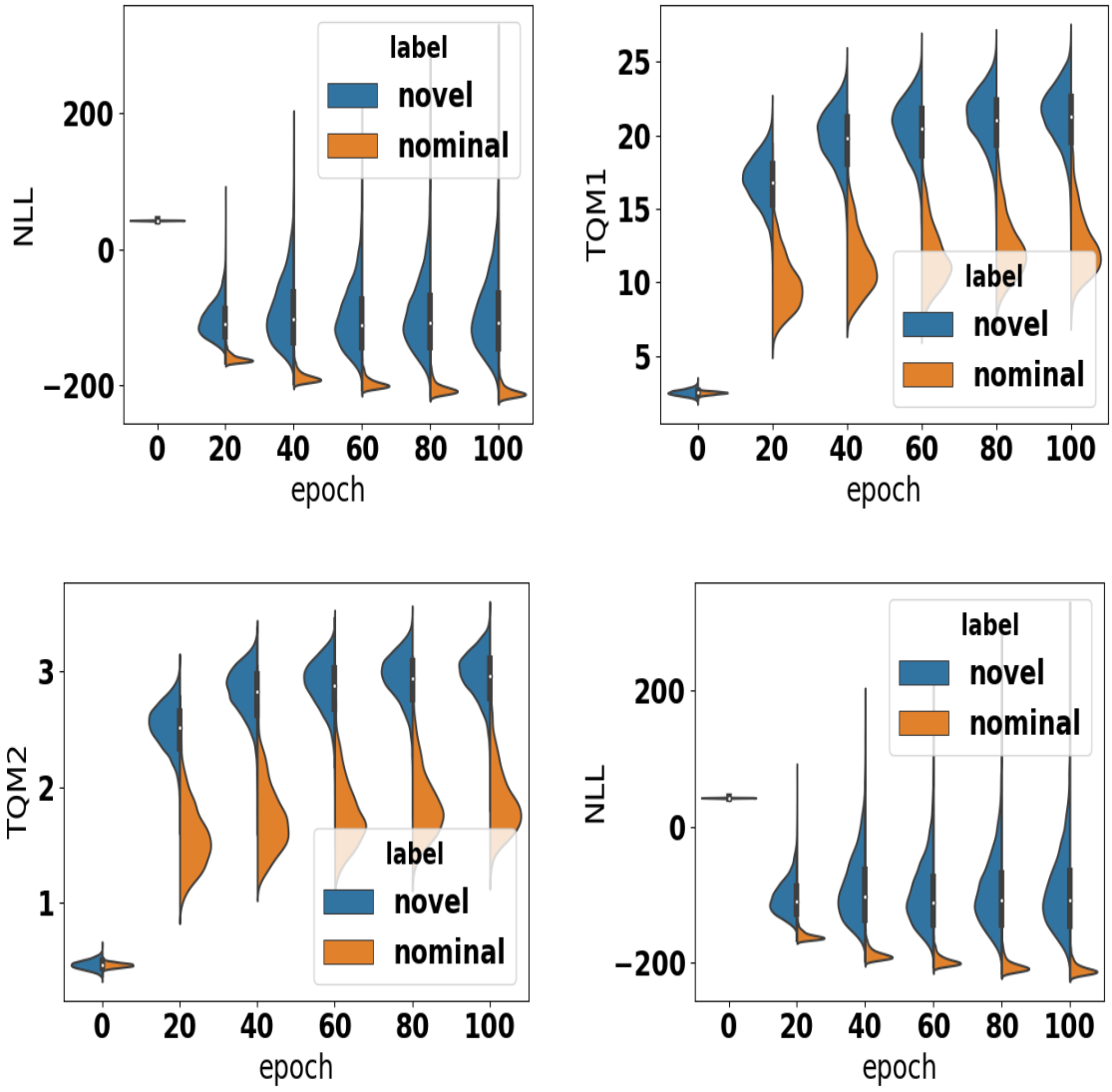

Figure 4.3: Distributional comparison on training and test scoring statistics on MNIST (nominal: digit 1). From left to right: 1) NLL; 2) $\texttt{TQM}_1$; 3) $\texttt{TQM}_2$; and 4) $\texttt{TQM}_\infty$.

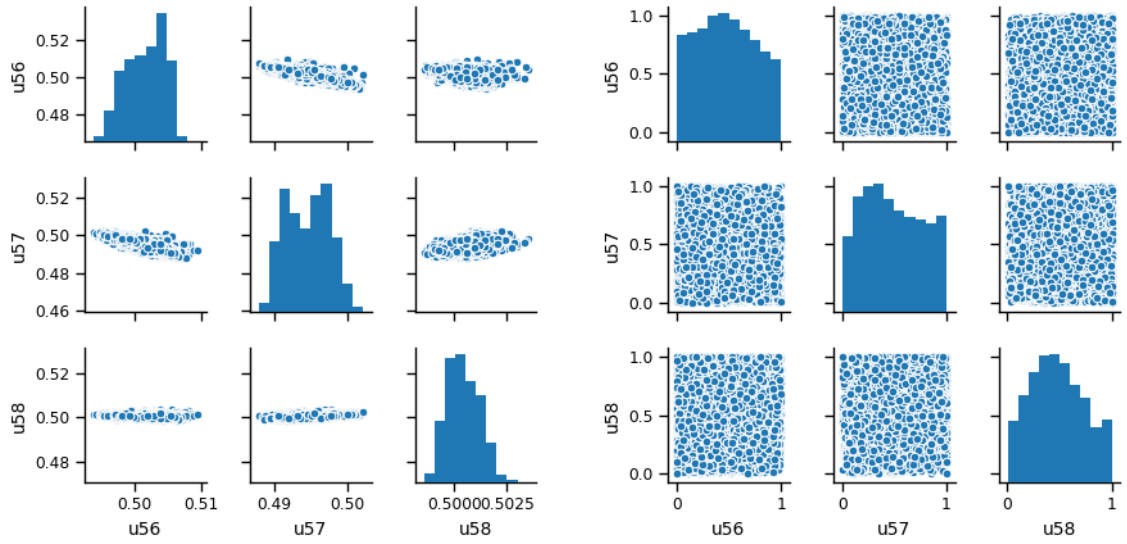

Figure 4.4: Marginal and joint distributions of pre-image in $[0, 1]$ of training data (dimension: $56, 57$, and 58). 1) distributions at initialization; and 2) distributions at 1000 epochs of training.

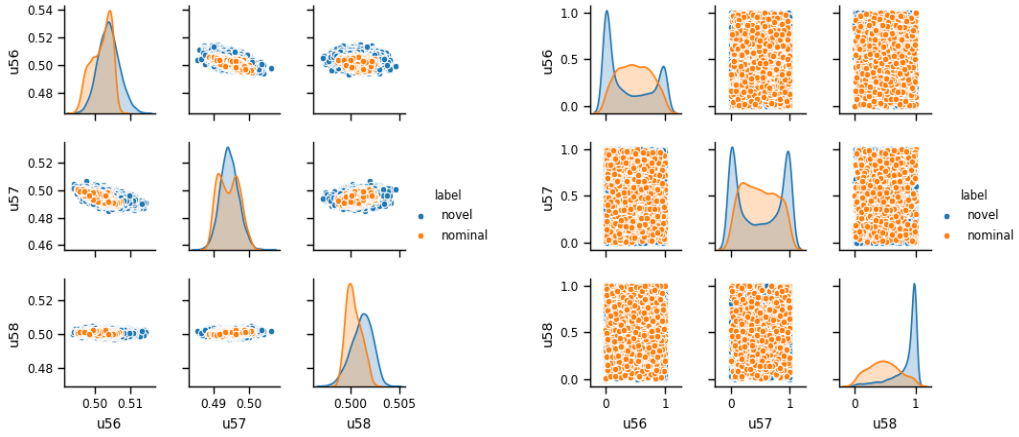

Figure 4.5: Marginal and joint distributions of pre-image in $[0, 1]$ of test data (dimension: $56, 57$, and 58). 1) distributions at initialization; and 2) distributions at 1000 epochs of training.

Chapter 5

Conclusion

In this thesis, we have developed a general framework for neural novelty detection, motivated by extending the univariate quantile function to the multivariate setting through increasing triangular maps.

Our framework unifies and extends many existing algorithms in novelty detection. We adapted the multiple gradient algorithm to obtain an efficient, end-to-end implementation of our framework that is free of any tuning hyperparameters. We performed extensive experiments on a number of datasets to confirm the competitiveness of our method against state-of-the-art alternatives.

References

[1] Jingjing Wang, Sun Sun, and Yaoliang Yu. Multivariate triangular quantile maps for novelty detection. *Advances in Neural Information Processing Systems*, 32, 2019.

[2] W. Fan, M. Miller, S. Stolfo, W. Lee, and P. Chan. Using artificial anomalies to detect unknown and known network intrusions. *Knowledge and Information Systems*, 6(5):507–527, 2004.

[3] Andrew B. Gardner, Abba M. Krieger, George Vachtsevanos, and Brian Litt. One-Class Novelty Detection for Seizure Analysis from Intracranial EEG. *Journal of Machine Learning Research*, 7:1025–1044, 2006.

[4] Paul M. Hayton, Bernhard Schölkopf, Lionel Tarassenko, and Paul Anuzis. Support Vector Novelty Detection Applied to Jet Engine Vibration Spectra. In *Advances in Neural Information Processing Systems 13*, pages 946–952, 2001.

[5] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. Regazzoni, and N. Sebe. Abnormal event detection in videos using generative adversarial nets. In *IEEE International Conference on Image Processing (ICIP)*, pages 1577–1581, 2017.

[6] M. Ravanbakhsh, E. Sangineto, M. Nabi, and N. Sebe. Training Adversarial Discriminators for Cross-Channel Abnormal Event Detection in Crowds. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1896–1904, 2019.

[7] W. Li, V. Mahadevan, and N. Vasconcelos. Anomaly Detection and Localization in Crowded Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):18–32, 2014.

[8] Mohammad Sabokrou, Mohsen Fayyaz, Mahmood Fathy, Zahra. Moayed, and Reinhard Klette. Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes. *Computer Vision and Image Understanding*, 172:88–97, 2018.

[9] Larry M. Manevitz and Malik Yousef. One-Class SVMs for Document Classification. *Journal of Machine Learning Research*, 2:139–154, 2001.

[10] Larry Manevitz and Malik Yousef. One-class document classification via Neural Networks. *Neurocomputing*, 70(7):1466–1481, 2007.

[11] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven Exploration by Self-supervised Prediction. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2778–2787, 2017.

[12] Markos Markou and Sameer Singh. Novelty detection: a review—part 1: statistical approaches. *Signal Processing*, 83(12):2481–2497, 2003.

[13] Markos Markou and Sameer Singh. Novelty detection: a review—part 2: neural network based approaches. *Signal Processing*, 83(12):2499–2521, 2003.

[14] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly Detection: A Survey. *ACM Computing Surveys*, 41(3):15:1–15:58, 2009.

[15] Marco A.F. Pimentel, David A. Clifton, Lei Clifton, and Lionel Tarassenko. A review of novelty detection. *Signal Processing*, 99:215–249, 2014.

[16] Shai Ben-David and Michael Lindenbaum. Learning Distributions by Their Density Levels: A Paradigm for Learning without a Teacher. *Journal of Computer and System Sciences*, 55(1):171–182, 1997.

[17] Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the Support of a High-Dimensional Distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[18] David M. J. Tax and Robert P. W. Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004.

[19] Ingo Steinwart, Don Hush, and Clint Scovel. A classification framework for anomaly detection. *Journal of Machine Learning Research*, 6:211–232, 2005.

[20] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun. Learning Discriminative Reconstructions for Unsupervised Outlier Removal. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1511–1519, 2015.

[21] Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. Deep Structured Energy Based Models for Anomaly Detection. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 1100–1109, 2016.

[22] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection. In *International Conference on Learning Representations*, 2018.

[23] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Georg Langs, and Ursula Schmidt-Erfurth. f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks. *Medical Image Analysis*, 54:30–44, 2019.

[24] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli. Adversarially Learned One-Class Classifier for Novelty Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3379–3388, 2018.

[25] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep One-Class Classification. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 4393–4402, 2018.

[26] Stanislav Pidhorskyi, Ranya Almohsen, and Gianfranco Doretto. Generative Probabilistic Novelty Detection with Adversarial Autoencoders. In *Advances in Neural Information Processing Systems 31*, pages 6823–6834, 2018.

[27] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks. In *International Conference on Learning Representations*, 2018.

[28] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. *Advances in neural information processing systems*, 31, 2018.

[29] Lucas Deecke, Robert Vandermeulen, Lukas Ruff, Stephan Mandt, and Marius Kloft. Image Anomaly Detection with Generative Adversarial Networks. In *Machine Learning and Knowledge Discovery in Databases*, pages 3–17, 2019.

[30] Sanjoy Dasgupta, Timothy C. Sheehan, Charles F. Stevens, and Saket Navlakha. A neural data structure for novelty detection. *Proceedings of the National Academy of Sciences*, 2018.

[31] Davide Abati, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Latent Space Autoregression for Novelty Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[32] Lukas Ruff, Robert A Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. *arXiv preprint arXiv:1906.02694*, 2019.

[33] Xuan Xia, Xizhou Pan, Nan Li, Xing He, Lin Ma, Xiaoguang Zhang, and Ning Ding. Gan-based anomaly detection: A review. *Neurocomputing*, 493:497–535, 2022.

[34] Lukas Ruff, Jacob R Kauffmann, Robert A Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G Dietterich, and Klaus-Robert Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 109(5):756–795, 2021.

[35] Priyank Jaini, Kira A. Selby, and Yaoliang Yu. Sum-of-Squares Polynomial Flow. In *Proceedings of The 36th International Conference on Machine Learning*, 2019.

[36] Aditya Krishna Menon and Robert C. Williamson. A loss framework for calibrated anomaly detection. In *Advances in Neural Information Processing Systems 31*, pages 1494–1504, 2018.

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

[38] H. Mukai. Algorithms for multicriterion optimization. *IEEE Transactions on Automatic Control*, 25(2):177–186, 1980.

[39] Jörg Fliege and Benar Fux Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.

[40] Jean-Antoine Désidéri. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5):313–318, 2012.

[41] Ivar Ekeland, Alfred Galichon, and Marc Henry. Comonotonic Measures of Multivariate Risks. *Mathematical Finance*, 22(1):109–132, 2012.

[42] Alfred Galichon and Marc Henry. Dual theory of choice with multivariate risks. *Journal of Economic Theory*, 147(4):1501–1516, 2012.

[43] Vladimir Igorevich Bogachev, Aleksandr Viktorovich Kolesnikov, and Kirill Vladimirovich Medvedev. Triangular transformations of measures. *Sbornik: Mathematics*, 196(3):309–335, 2005.

[44] G. L. O'Brien. The Comparison Method for Stochastic Processes. *The Annals of Probability*, 3(1):80–88, 1975.

[45] Elja Arjas and Tapani Lehtonen. Approximating Many Server Queues by Means of Single Server Queues. *Mathematics of Operations Research*, 3(3):205–223, 1978.

[46] Ludger Rüschendorf. Stochastically ordered distributions and monotonicity of the oc-function of sequential probability ratio tests. *Series Statistics*, 12(3):327–338, 1981.

[47] Alexis Decurninge. *Univariate and multivariate quantiles, probabilistic and statistical approaches; radar applications*. PhD thesis, Université Pierre et Marie Curie-Paris VI, 2015.

[48] Youssef Marzouk, Tarek Moselhy, Matthew Parno, and Alessio Spantini. *Sampling via Measure Transport: An Introduction*, pages 1–41. Springer, 2016.

[49] Alessio Spantini, Daniele Bigoni, and Youssef Marzouk. Inference via low-dimensional couplings. *Journal of Machine Learning Research*, 19:1–71, 2018.

[50] Cédric Villani. *Optimal Transport: Old and New*, volume 338. Springer, 2008.

[51] Victor Chernozhukov, Alfred Galichon, Marc Hallin, and Marc Henry. Monge–Kantorovich depth, quantiles, ranks and signs. *The Annals of Statistics*, 45(1):223–256, 2017.

[52] David Inouye and Pradeep Ravikumar. Deep Density Destructors. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2167–2175, 2018.

[53] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

[54] Meshi Bashari, Amir Epstein, Yaniv Romano, and Matteo Sesia. Derandomized novelty detection with fdr control via conformal e-values. *Advances in Neural Information Processing Systems*, 36, 2024.

[55] Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural Autoregressive Flows. In *ICML*, 2018.

[56] Akiko Takeda and Masashi Sugiyama. ν-Support Vector Machine as Conditional Value-at-Risk Minimization. In 25^{th} *International Conference on Machine Learning*, pages 1056–1063, 2008.

[57] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360*, 2018.

[58] Andrew Barto, Marco Mirolli, and Gianluca Baldassarre. Novelty or Surprise? *Frontiers in Psychology*, 4:907, 2013.

[59] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*, 2014.

[60] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[61] Yann LeCun. The mnist database of handwritten digits. http://yann.lecun.com/exdb/mnist/.

[62] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747, 2017.

[63] Moshe Lichman. UCI machine learning repository. http://kdd.ics.uci.edu/databases/kddcup99.

[64] Moshe Lichman. UCI machine learning repository. http://archive.ics.uci.edu/ml.

[65] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pages 2338–2347, 2017.

APPENDICES

Appendix A

Datasets and network architectures

In this section, we briefly describe the datasets, the network architectures, as well as the hyper-parameters that are used in our proposed algorithm. For all image datasets, the pixel values of each image are scaled to $[0, 1]$. For non-image datasets, no extra preprocessing is applied. The statistics of the datasets are summarized in Table A.1.

- **MNIST** [61]
  - Dataset description: MNIST [61] includes $70,000$ grayscale images of numeric digits from 0 to 9, each of size 28×28. There are $7,000$ examples per class. The training set contains $60,000$ examples, and the test set contains $10,000$ examples.
  - Network architecture: We use the same autoencoder as that in `LSA`, and the dimension of the latent vector is set to 64. The estimation network is based on SOS [35], which contains multiple blocks each consisting of a SOS-flow layer, a normalization flow layer and a reversing layer. The number of blocks is set to 1. In the SOS-flow layer, we set $k = 5$ and $r = 4$ in Eqn.(8) in [35]. All the parameters are generated by a conditioner network, which contains one fully-connected layer: FC(724, 64, none)-FC(64, c, none), where c is the number of the parameters in the SOS-flow layer.
  - Optimization hyperparameters: The number of epochs is set to 1000, and training is stopped after 100 epochs of non-decreasing loss. The size of each mini-batch is 256. We use Adam with the learning rate 10^{-5}.
- **Fashion-MNIST** [62]

- Dataset description: Fashion-MNIST includes $70,000$ grayscale images of fashion products in 10 classes. This dataset has the same image size and the structure of training and test splits as in MNIST.
- Network architecture: same as that in MNIST.
- Optimization hyperparameters: same as those in MNIST.

- **KDDCUP** [63]
  - Dataset description: KDDCUP dataset contains 125 dimensions in total. In this dataset, 20% of data are labeled as "normal" and the rest are labeled as "attack". We treat "normal" data as novel since they are minority.
  - Network architecture: We use the same autoencoder as that in `DAGMM` except that the dimension of the latent vector is set to 2. The structure of the autoencdoer is as follows: FC(125,60,tanh)-FC(60,30,tanh)-FC(60,30,tanh)-FC(30,10,tanh)-FC(10,2,none)-FC(2,10,tanh)-FC(10,30,tanh)-FC(30,60,tanh)-FC(60,125,tanh).
  - Optimization hyperparameters: The size of each mini-batch is 1024. The learning rate in Adam is 10^{-5}. Training is stopped after 100 epochs of non-decreasing loss.

- **Thyroid** [64]
  - Dataset description: Thyroid dataset consists of three classes. We treat the hyperfunction class as the novel class and the rest as the nominal class.
  - Network architecture: We remove the autoencoder and only use the same estimation network as that in MNIST.
  - Optimization hyperparameters: The size of each mini-batch is 1024. The learning rate in Adam is 10^{-3}. Training is stopped after 100 epochs of non-decreasing loss.

Table A.1: Statistics of Datasets

	Dimension	Instance	Classes	Anomaly ratio
MNIST	784	70,000	10	0.9
Fashion-MNIST	784	70,000	10	0.9
KDDCUP	125	494,021	2	0.2
Thyroid	6	3,772	2	0.025

Appendix B

Comparison between density and quantile approaches

Theorem 1 *In the univariate case, if the nominal distribution F_0 is unimodal and symmetric w.r.t the origin, then the density approach and the quantile approach achieve the same* AUROC.

Proof: It is well-known that AUROC is equal to the probability of a random nominal example being ranked higher than a random novel example, i.e.,

$$\text{AUROC} = \Pr(\mathsf{S}(X_0) > \mathsf{S}(X_1)), \tag{B.1}$$

where $X_0 \sim F_0$ is nominal and $X_1 \sim F_1$ is novel, and S is the scoring rule.

For the density approach, we have $\mathsf{S} = f_0$, where $f_0 = F_0'$ is the density of the nominal distribution. Thus,

$$\text{AUROC}_{\text{NLL}} = \Pr(f_0(X_0) > f_0(X_1)) = \Pr(|X_0| < |X_1|), \tag{B.2}$$

where the last equality follows from the unimodal and symmetric assumption on f_0.

On the other hand, for the quantile approach, the scoring rule is $\mathsf{S} = -|F_0 - \frac{1}{2}|$ (note the negation since we assume the higher S is the more nominal it is). Thus, where again the last equality is due to the unimodal and symmetric assumption on F_0. ■

Remark 3 *There is nothing special about the origin: the same result holds if F_0 is unimodal and symmetric w.r.t any point c.*

Remark 4 *We suspect a similar result holds for multivariate distributions as well. A natural condition on f_0 is that its contours are multiples of the ℓ_∞ ball. We need to show that the TQM for such distributions are symmetric in some sense.*

Theorem 2 *In the univariate case, if the nominal distribution F_0 is uni-modal and symmetric, then the density approach and the quantile approach lead to the same ROC curve.*

Proof: Denote the novel data as positive and nominal data as negative. For the quantile approach, given a threshold t_q the set of data identified as novel can be characterized by $\{x : |F_0(x) - \frac{1}{2}| > t_q\}$. In contrast, for the density approach, given a threshold t_d the identified novel data can be characterized by $\{x : -f_0(x) > t_d\}$, where $f_0(x)$ is the density of the nominal distribution. For both cases, the left hand side of the inequality represents the scoring function, and the higher the value of the scoring function the more likely the data being identified as novel.

In an ROC curve, each point is associated with a threshold. Therefore, to prove the result it suffices to show that there exists a one-to-one correspondence between t_q and t_d that leads to the same partition of the novel and nominal regions under the quantile and density approach respectively. Obviously, if F_0 is uni-model and symmetric, given t_q we can set $t_d = -f_0(F^{-1}(t_q + \frac{1}{2}))$ and the partition is the same. ■

Remark 5 *In general the above conclusion cannot be extended to the multivariate case. For example, assume that the nominal data follows the 2-D standard Gaussian. Then, under the density approach, the boundary between novel and nominal data is an ellipsoid; while under the quantile approach, the boundary is square (assuming we employ the infinity norm scoring rule). The corresponding experimental results are shown below.*

B.1 1-D: uni-model and symmetric model

Assume that the nominal data follows the standard univariate Gaussian distribution $N(10, 1)$. Consider two types of novel data: I) novel data are far away from nominal data, say following $N(15, 1)$; and II) novel data are near nominal data, say following $N(12, 1)$. For both cases, the density and quantile methods have exactly the same ROC curve, confirming our theoretical results above. In particular, for the first case, the curve goes vertically from $(0, 0)$ to $(0, 1)$, and then horizontally to $(1, 1)$, indicating perfect performance in anomaly detection. See Figures B.1 and B.2.

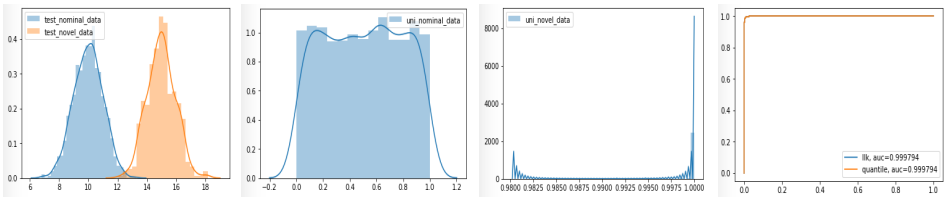

Figure B.1: Type I novel data. 1) distribution of test data; 2) distribution of pre-image in $[0, 1]$ for nominal data; 3) distribution of pre-image in $[0, 1]$ for novel data; and 4) ROC curve.

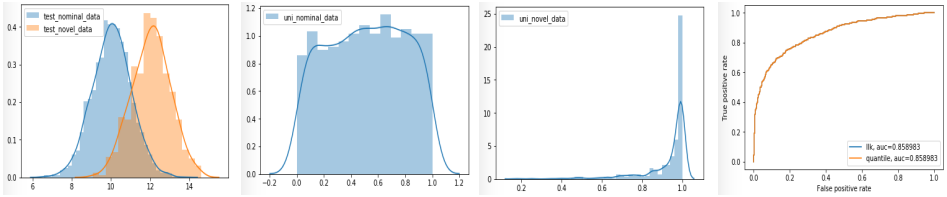

Figure B.2: Type II novel data. 1) distribution of test data; 2) distribution of pre-image in $[0, 1]$ for nominal data; 3) distribution of pre-image in $[0, 1]$ for novel data; and 4) ROC curve.

B.2 1-D: mixture model

Assume that the nominal data follows the Gaussian mixture model $0.7N(0, 2^2) + 0.3N(10, 1)$. Consider two types of novel data: I) novel data is far away from nominal data, say following $N(15, 1)$; and II) novel data is surrounded by nominal data, say following $N(5, 1)$. For the first case, both methods have perfect performance; while for the second case, the quantile method is dominated by the density method. See Figures B.3 and B.4.

B.3 2-D: uni-modal and symmetric model

Assume that the nominal data follows the 2-D Gaussian distribution with mean $[0, 0]$ and covariance matrix $[1, 0; 0, 1]$. Consider two types of novel data: I) novel data is far away from nominal data, say following the 2-D Gaussian distribution with mean $[5, 5]$ and covariance matrix $[1, 0; 0, 1]$; and II) novel data is near nominal data and follows the 2-D Gaussian distribution with mean $[2, 2]$ and covariance matrix $[1, 0; 0, 1]$. For the first case, both methods have perfect performance; while for the second case, the density method is slightly better than the quantile method. See Figures B.5 and B.6.

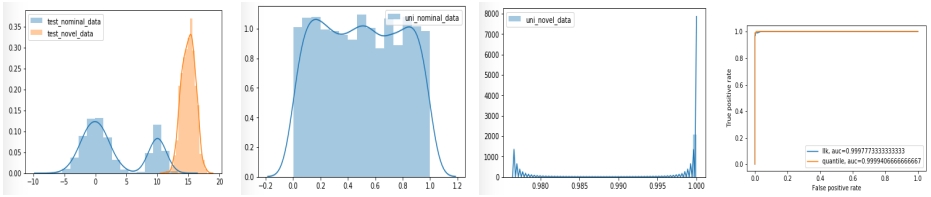

Figure B.3: Type I novel data. 1) distribution of test data; 2) distribution of pre-image in $[0, 1]$ for nominal data; 3) distribution of pre-image in $[0, 1]$ for novel data; and 4) ROC curve.

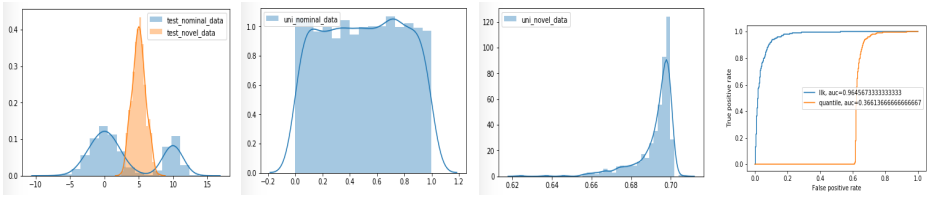

Figure B.4: Type II novel data. 1) distribution of test data; 2) distribution of pre-image in $[0, 1]$ for nominal data; 3) distribution of pre-image in $[0, 1]$ for novel data. The novel data are projected around the middle instead of at the ends; and 4) ROC curve.

B.3.1 2-D: donut example

Let us consider the donut distribution[1]

$$p(x, y) = \begin{cases} \frac{1}{3\pi}, & \text{if } 1 \leq x^2 + y^2 \leq 4 \\ 0, & \text{otherwise} \end{cases} . \tag{B.3}$$

Under the increasing triangular map $\mathbf{Q}$, the pre-images of x and y in $[0, 1]^2$ are $F(x)$ and $F(y|x)$, respectively, where $F(\cdot)$ denotes the cumulative distribution function.

The marginal density of x can be represented by

$$p(x) = \begin{cases} \frac{2}{3\pi}(\sqrt{4 - x^2} - \sqrt{1 - x^2}), & \text{if } -1 < x < 1 \\ \frac{2}{3\pi}\sqrt{4 - x^2}, & \text{if } -2 \leq x \leq -1 \text{ or } 1 \leq x \leq 2 \\ 0 & \text{otherwise} \end{cases} . \tag{B.4}$$

[1]We thank an anonymous reviewer for suggesting this example.

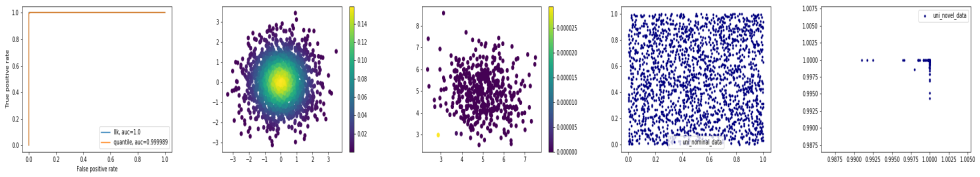

Figure B.5: Type I novel data. 1) ROC curve; 2) density of nominal test data; 3) density of novel test data; 4) pre-image of nominal data in $[0,1]^2$; and 5) pre-image of novel data in $[0,1]^2$.

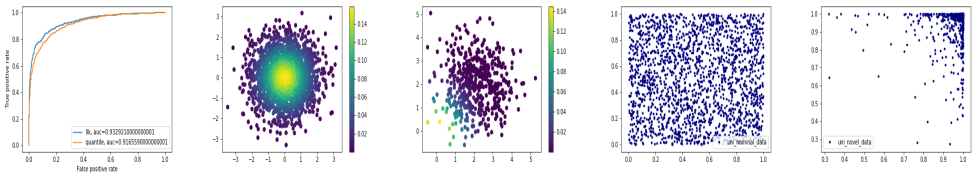

Figure B.6: Type II novel data. 1) ROC curve; 2) density of nominal test data; 3) density of novel test data; 4) pre-image of nominal data in $[0,1]^2$; and 5) pre-image of novel data in $[0,1]^2$.

Then $F(x)$ can be calculated as follows:

$$F(x) = \begin{cases} 0, & \text{if } x < -2 \\ \frac{2}{3\pi}\left(\frac{x}{2}\sqrt{4-x^2} + 2\arcsin\frac{x}{2} + \pi\right), & \text{if } -2 \le x < -1 \\ \frac{2}{3\pi}\left(\frac{3}{4}\pi + \frac{x}{2}\sqrt{4-x^2} - \frac{x}{2}\sqrt{1-x^2} + 2\arcsin\frac{x}{2} - \frac{1}{2}\arcsin x\right), & \text{if } -1 \le x < 1 \\ \frac{2}{3\pi}\left(\frac{\pi}{2} + \frac{x}{2}\sqrt{4-x^2} + 2\arcsin\frac{x}{2}\right), & \text{if } 1 \le x < 2 \\ 1, & \text{otherwise} \end{cases} . \tag{B.5}$$

Given x, y is uniformly distributed.

1. If $-2 \le x \le -1$ or $1 \le x \le 2$, the conditional density $p(y|x)$ can be represented by

$$p(y|x) = \begin{cases} \frac{1}{2\sqrt{4-x^2}}, & \text{if } -\sqrt{4-x^2} \le y \le \sqrt{4-x^2} \\ 0, & \text{otherwise} \end{cases}, \tag{B.6}$$

and the corresponding conditional CDF

$$F(y|x) = \begin{cases} 0, & \text{if } y < -\sqrt{4-x^2} \\ \frac{y+\sqrt{4-x^2}}{2\sqrt{4-x^2}}, & \text{if } -\sqrt{4-x^2} \le y < \sqrt{4-x^2} \\ 1, & \text{otherwise} \end{cases}. \tag{B.7}$$

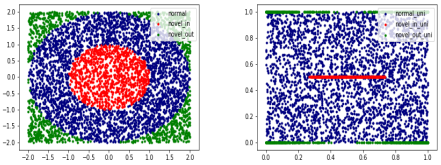

Figure B.7: Donut example: 1) samples of nominal and two types of novel data; and 2) analytical pre-images in $[0, 1]^2$ for nominal and novel data.

2. If $-1 < x < 1$,

$$p(y|x) = \begin{cases} \frac{1}{2(\sqrt{4-x^2}-\sqrt{1-x^2})}, & \text{if } -\sqrt{4-x^2} \le y \le -\sqrt{1-x^2} \text{ or } \sqrt{1-x^2} \le y \le \sqrt{4-x^2} \\ 0, & \text{otherwise} \end{cases}, \tag{B.8}$$

and the corresponding conditional CDF

$$F(y|x) = \begin{cases} 0, & \text{if } y < -\sqrt{4-x^2} \\ \frac{y+\sqrt{4-x^2}}{2(\sqrt{4-x^2}-\sqrt{1-x^2})}, & \text{if } -\sqrt{4-x^2} \le y < -\sqrt{1-x^2} \\ \frac{1}{2}, & \text{if } -\sqrt{1-x^2} \le y < \sqrt{1-x^2} \\ \frac{1}{2} + \frac{y-\sqrt{1-x^2}}{2(\sqrt{4-x^2}-\sqrt{1-x^2})}, & \text{if } \sqrt{1-x^2} \le y < \sqrt{4-x^2} \\ 1, & \text{otherwise} \end{cases}. \tag{B.9}$$

On Figure B.7 (left) we show the random samples of the nominal and novel data, and on Figure B.7 (right) we show the pre-images of these samples in the square $[0, 1]^2$ using the derived analytical formula. It can be seen that the outer novel data is projected onto the boundary of the square hence can be identified using $\texttt{TQM}_\infty$. The inner novel data, however, cannot be identified easily using the current quantile-based scoring functions. To improve the performance we might need some prior knowledge of such novel data and then adjust the scoring function accordingly. In contrast, the density approach would work well by setting a density threshold between 0 and $\frac{1}{3\pi}$.

In Figure B.8, instead of applying the analytical formula we use an SOS-based estimation network to learn the TQM. The observation is generally consistent with that derived using the analytical formula.

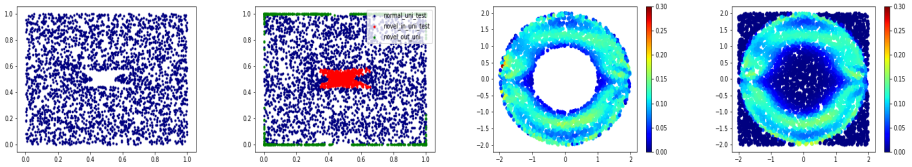

Figure B.8: Donut example: The density and the quantile map are learned by SOS. (1) pre-image of nominal training data in $[0,1]^2$; 2) pre-image of test data in $[0,1]^2$; 3) density of nominal training data; and 4) density of test data.

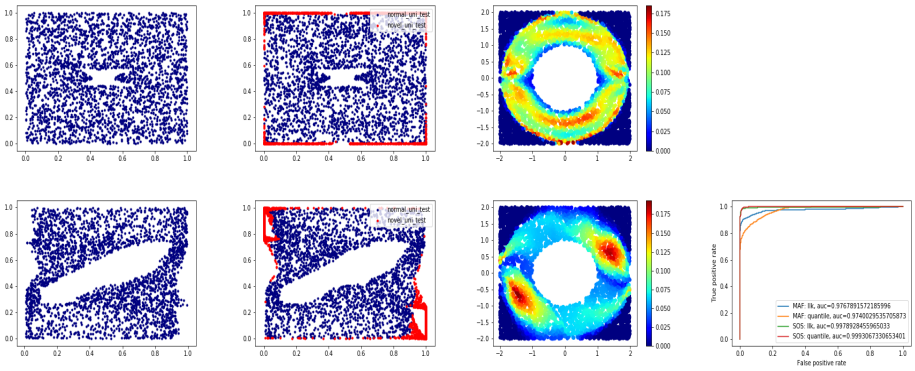

Figure B.9: Donut example: In the first row, the density and the quantile map are learned by SOS: (1) pre-image of nominal training data in $[0,1]^2$; 2) pre-image of test data in $[0,1]^2$; and 3) density of test data. In the second row, the density and the quantile map are learned by MAF. The last plot shows the comparison of ROC curves.

B.3.2 Discussion

The quantile and density methods apply different scoring functions to identify novel data. Specifically, under the density method data with a low density (or log-likelihood) is deemed novel, while under the quantile method, for example, data projected to the boundary regions of the hypercube $[0,1]^d$ is deemed novel. A main advantage of the quantile method is that by checking whether data projected onto $[0,1]^d$ is uniformly distributed we can tell whether the quantile map is estimated successfully. In contrast, for the density method, generally it is difficult to assess the accuracy of the estimated density. To give an example, consider the donut example in Section B.3.1 and assume the outer data as novel. In Figure B.9, we show the results when the TQM $\mathbf{Q}$ is learned by SOS and MAF [65], respectively. By projecting data onto $[0,1]^2$ (using the inverse TQM), we can conclude that SOS learns a better quantile map and indeed the corresponding ROC curve

dominates that under MAF.

We also point out that under the current quantile thresholding rules (see §4.1.1) the identified nominal region is generally (path) connected, due to the increasing requirement we impose on TQM. Therefore, provided that nominal data follows some multi-modal distribution and novel data is located between different modes, as the shown example of 1-D mixture model in §B.2, the current quantile scoring rules would not work well. This reveals the importance of learning a (unimodal) hidden representation in our framework (2.3). It would be interesting to design new quantile thresholding rules to induce disconnected nominal region.