Disentangled Conditional Variational Autoencoder for Unsupervised Anomaly Detection

by

Asif Ahmed Neloy

A Thesis submitted to the Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science
University of Manitoba
Winnipeg

Dedication

وَاصْبِرْ وَمَا صَبْرُكَ إِلاَّ بِاللَّهِ وَلاَ تَحْزَنْ عَلَيْهِمْ وَلاَ تَكُ فِي ضَيْقٍ مِمَّا يَمْكُرُونَ ﴿النحل

“ - And bear with patience, and your patience is only because of the help of Allah and do not grieve over them, nor feel distressed by their evil plans”

—Al Quran, (16:127)

This is dedicated to my ***mother*** and the ones I *love*

Acknowledgements

I would like to convey my sincere gratitude and thank my supervisor Dr. Maxime Turgeon and my co-supervisor, Dr. Cuneyt Akcora. It was my great pleasure and honor to work under their supervision. Over the years, Dr. Turgeon helped me with the knowledge, keen guidance, and support I needed to reach this stage. Back in 2019, when Dr. Turgeon and I had our first meeting, I was totally uncertain and afraid of stepping into a new country, staying thousands of miles away from home, and embracing the new life amidst the pandemic. Dr. Turgeon was incredibly patient and amiable throughout my journey, guiding me with all the resources and knowledge that I needed to step into the world of mathematics, whereas I was too afraid to derive $x^n\,dx$. All his wisdom, key attention, and thoughtful comments and observations were truly instrumental to my career and helped me to grow as a researcher.

I want to acknowledge the financial support I received from the NSERC CREATE grant on the Visual and Automated Disease Analytics (VADA) program, Department of Computer Science, Faculty of Graduate Studies, and Faculty of Science, University of Manitoba.

I am truly grateful to my family, especially my mother, the love of my life, who always believed in me, and encouraged me to reach where I am today. Despite having severe depression, and sickness, I conquered my journey with their support. Not to mention, my thesis would not have been completed without DKP.

Abstract

The goal of efficient anomaly or outlier detection is to learn the hidden representation of the data by identifying independent factors and minimizing information loss. Variational Autoencoder (VAE) and its extensions have shown great promise to learn the data distribution. In this manuscript-based thesis, I propose a novel architecture of generative framework to investigate the following objectives: (a) Effectively learn disentangled representations of data by optimizing total correlation (TC) loss. (b) Minimize information loss using mutual information theory for better reconstruction ability and appropriate sample generation. (c) Address sample reconstruction error vs. reconstruction quality trade-offs. In the first manuscript, I review the architectures of autoencoders divided into three main categories: classical, variational, and regularized autoencoders. Then, I present their mathematical foundation and explore their ability to detect anomalies, reconstruct samples and learn latent factors. In the second manuscript, I propose ***D***isentangled ***C***onditional ***V***ariational ***A***utoencoder ***(dCVAE)***, which combines the frameworks of β-VAE, conditional variational autoencoder (CVAE), and the principle of total correlation (CorEx). Through experiments, I show that the accuracy of anomaly detection methods can be improved while learning disentangled factors and minimizing information loss. This can be done by connecting multivariate information theory and regularizing the posterior-variant of VAE. Finally, I conclude this thesis by discussing limitations, and I give a brief overview of future research directions for VAE architectures in high-dimensional image datasets.

Contents

List of Figures

List of Tables

Acronyms

AAE	Adversarial Autoencoder
AD	Anomaly Detection
adVAE	Self-Adversarial Variational Autoencoder
AP	Average Precision
CAE	Contractive Autoencoder
CorEx	The Principle of Total Correlation
CorGAN	Corrupted Generative Adversarial Networks
CVAE	Conditional Variational Autoencoder
DAE	Denoising Autoencoder
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
dCVAE	Disentangled Conditional Variational Autoencoder
DT	Decision Tree
eForest	EncoderForest
ELBO	Evidence Lower Bound
EM	Expectation-Maximization
GAN	Generative Adversarial Networks
GMM	Gaussian Mixture Model
IWAE	Importance Weighted Autoencoder
KLD	Kullback-Leibler Divergence
LLE	Locally Linear Embedding
LOF	Local Outlier Factor
MDS	Multi-dimensional Scaling
MIG	Mutual Information Gap
NF	Normalizing Flow
NN	Neural Networks
ocSVM	One-Class Support Vector Machine

OD	Outlier Detection
PAE	Probabilistic Autoencoder
PCA	Principal Component Analysis
RDA	Robust Deep Autoencoder
RNN	Recurrent Neural Networks
ROC	Receiver operator characteristic
RPCA	Robust Principal Component Analysis
SAD	Supervised Anomaly Detection
SAE	Sparse Autoencoder
scAE	Stacked Convolutional Autoencoder
SGVB	Stochastic Gradient Variational Bayes
SSAD	Semi-Supervised Anomaly Detection
STN	Spatial Transformer Networks
SVDD	Support vector data description
SVM	Support Vector Machine
t-SNE	t-distributed Stochastic Neighbor Embedding
TC	Total Correlation
UAD	Unsupervised Anomaly Detection
VAE	Variational Autoencoder
VQ-VAE	Vector Quantized Variational Autoencoder

Chapter 1

Introduction

Over the last decade, data sources are becoming more prominent and ubiquitous. As a result, anomalous data points now present themselves in a more complex and various ways. Due to the volume of data, the nature of anomalous points changes over time and heavily depends on the context of the application. As a result, anomaly detection (AD) or outlier detection (OD) can prove essential to many applications and is considered a non-trivial task. Moreover, benign or erroneous AD compromises the data quality and integrity, which leads to ineffective data analysis or implementation of pattern recognition techniques.

Anomaly or outlier detection has imposed several challenges over the year. Detecting such anomalous and corrupted data, understanding the high-dimensional complex data, and effectively decoding it into a convenient information set are considered primary goals of unsupervised statistical deep learning frameworks. The contributions of deep learning models for anomaly detection spread over several applications over the years, including but not limited to: data cleansing, computational biology, ecosystem disturbance detection, medicine, clinical trials, financial transactions, weblogs, telecommunications, geographic information systems, network intrusion, and fault/damage detection. More recently, unsupervised anomaly detection (UAD) methods have emerged as one of the most promising approaches to achieving breakthrough performance on different tasks.

The downstream tasks of UAD are non-trivial and heavily depend on the appropriate definition that is often problem specific. The high dimensionality of image data creates difficulties for UAD methods due to the diverse number of attributes

or features, the amount of available data, and the increasing sparsity in data. Data sparsity generally rises due to unnecessary factors and variables, multicollinearity among features, irrelevant attributes, and high noise levels. This issue is extensively acknowledged as the *curse of dimensionality*. Prior to using generative and variational deep learning approaches, including Variational Autoencoder (VAE) and Generative Adversarial Networks (GAN), heuristic statistical methods were the primary tools for UAD in restricted application domains.

Current VAEs and its variants primarily focus on the following four directions:

- Improving the quality of reconstruction or image generation.
- Enhancing the disentanglement for VAEs.
- Enforcing regularization to address trade-offs between efficient training and model loss.
- Developing Bayesian models using prior-variant or prior-variance based on the data distribution.

VAE poses solid theoretical background to control the distribution of the latent representation, learn the smooth latent representations of the input data, and generating new meaningful samples. However, VAE struggles to disentangle highly correlated high-dimensional data. *Disentanglement learning* is defined as the identification of independent or uncorrelated representations of high-dimensional data. The idea of learning disentangled representations is to map the high-dimensional input data to a lower-dimensional representation such that the original input or code can be approximately reconstructed without losing valuable information. Although recently proposed deep learning architectures such as β-VAE [2], β-TCVAE [3], Factor VAE [4], InfoVAE [5], VFAE [6], and DIP-VAE [7] utilized the architecture of VAE to address the disentangled factors issues, determining how effective such learned representation is for UAD task is not yet well understood. Hence, it is critical to understand which VAE backbone is appropriate for learning disentanglement and how generic mutual information theories are useful in enforcing losing valuable information in such settings.

Additionally, maximizing the reproducibility and minimizing the computational cost is still an active research area yet to be adequately addressed for such deep

learning frameworks. The tasks of detecting anomalies on real-world image datasets vs. generated synthetic image datasets are distinct. Real-world datasets exhibit more prominent variability due to the heterogeneity in abnormality presentation across data points or cohorts and differences in acquisition colors or encoding features. Anomalies in such datasets tend to have a finer resolution or more localized features that are incredibly challenging to distinguish. Using representation learning to learn disentangled factors in a pixel-wise image is widely adopted and exercised [8]. However, utilizing the disentangled factors to detect anomalies in a whole image is a challenge that has yet to be addressed appropriately. Moreover, an unsupervised deep learning framework heavily depends on the number of features and artifacts of a dataset (such as color, boundaries, strokes, and size) and often leads to losing valuable attributes. Thus, learning disentangled factors and minimizing information loss is a highly challenging UAD task in the context of image data.

1.1 Contributions

The immediate contribution of this thesis is a new architecture of generative model based on multivariate mutual information theory combined with a conditional VAE architecture for the UAD task. The technique is tailored to image data type and can be applied to various image sets. The other contributions of this thesis follow.

In Chapter 3, "A Comprehensive Study of Autoencoders for Anomaly Detection: Efficiency and Trade-Offs," I present an overview of the architectures and theoretical foundation of autoencoders. Through Experiments, the baseline comparison method for different autoencoders is identified and ranked based on their overall performance. Then, I Perceived the trade-offs between reconstruction error and reconstruction quality and proposed a method to learn disentangled representation. Finally, this review concludes with the pros and cons of different VAE architectures and proposes future research directions.

In Chapter 4, the capacity of a disentangled representation learning for VAE is comprehensively explored, which includes conditional generations (CVAE) and the principle of total correlation (CorEx) as a reference implementation. The potential of CVAE in combination with VAE and CorEx for AD is thoroughly investigated by

learning efficient disentangled factors to minimize information loss. To the best of my knowledge, this is the first attempt to explore disentangled learning and multivariate information theory in anomaly detection. Afterward, my proposed method is evaluated extensively on a range of image data, including MNIST, Fashion-MNIST, EMNIST, and KMNIST, for a better understanding of the results tested in real-world datasets. Finally, through several experimental techniques, such as inspecting latent variables, sample reconstruction, and evaluating accuracy metrics, I pursued the widely acknowledged challenge of "*reconstruction vs. loss*" trade-off. By proposing a new objective function, I provide empirical evidence to minimize such trade-offs effectively.

1.2 Thesis Outline

In this thesis, I consider the problem of UAD with autoencoders and propose a new architecture of variational autoencoder techniques that enables a generative model to learn useful disentangled representations of the input data and effectively perform AD in an unsupervised fashion.

The thesis is structured as follows: in Chapter 2, I review the background and literature on UAD methods. The reviewed methods primarily focus on VAE architectures and their variants, applied to image data. I provide a brief overview of UAD frameworks that includes variants of VAE, multivariate information theories, meta-priors, and emphasis applications of AD. This chapter concludes by providing a connection between VAE architectures and information theory.

The following two chapters present the original contributions from two manuscripts. Chapter 3 briefly reviews different autoencoder methods, provides an overview of their architecture, and concise experimental observations to audit the task of UAD. It also includes future research directions for VAE architectures. Then, Chapter 4 introduces my proposed Disentangled Conditional Variational Autoencoder for Unsupervised Anomaly Detection (dCVAE) framework. In that chapter, I provide the background studies, several VAE approaches, and the mathematical background of the newly proposed objective function. Additionally, I explain the details of dCVAE's experimental design, discuss the results, and analyze the framework's

performance. Through several evaluation tasks, I show how dCVAE leads to better results and improves the downstream tasks of UAD compared to common VAE architectures. Finally, I conclude my thesis (Chapter 5) by summarizing the contributions, and by discussing the limitations and future research directions.

Chapter 2

Literature Review

As discussed in the Chapter 1, this thesis focuses on unsupervised anomaly detection and perceiving the relationship between sample reconstruction and minimizing reconstruction loss. More specifically, I look forward to acquiring empirical evidence of learning efficient anomaly detection and disentangled factors' effects for efficient loss function minimization. Therefore, I focus on the following two topics in this chapter: variations of VAE architectures, including regularizing posterior-variant, and conditional generations, utilizing total correlation, and learning meta-priors. Firstly, I define different types of anomalies and provide an overview of anomaly detection methods (Section 2.1). Later, I review the taxonomy of anomaly detection methods (Section 2.2). Finally, in Section 2.3, a comparison between different UADs is characterized, followed by the architectures of autoencoders. This chapter concludes by briefly summarizing the association between different VAE architectures.

2.1 Anomaly and Anomaly Detection

Gladitz [9] defines an outlying observation, or outlier, as an observations that deviates markedly from other sample observations. Johnson et al. [10] defines an *outlier* as an observation in a data set that appears inconsistent with the remainder of that data set. In statistical regression, *anomalies* or *outliers* are observations that derive abnormal distances from other instances in a random sample from a population. In general, identifying or detecting such observations is known as

outlier or anomaly detection. Chandola et al. [11] defined AD as "the problem of finding patterns in data that do not conform to expected behavior." Outlier or novelty detection are synonyms for AD.

2.2 Classification of Anomalies

In this section, a brief classification of anomaly types is presented. Through this classification, the latter sections are organized to identify and extensively evaluate the different models and architectures for detecting anomalies.

2.2.1 Nature of the Input Data

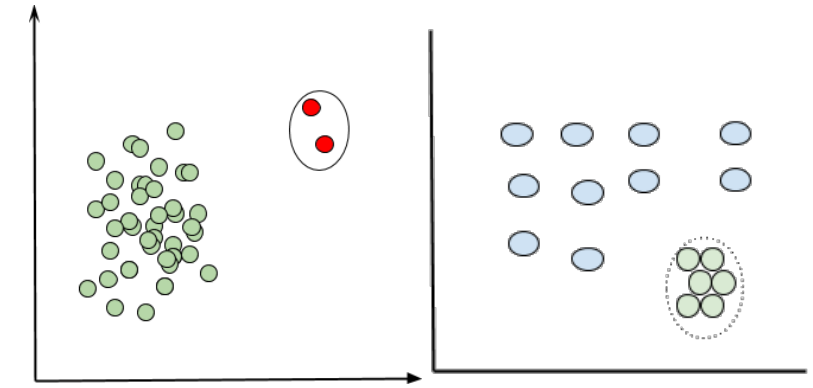
In general, based on the nature of the input data, the anomalies are classified into three categories:

1. **Point Anomalies**
   Point anomalies or global anomalies represent the deviation or irregularity of a specific data point without any pattern or interpretation compared to the whole dataset. Point anomalies can be classified as a context anomalies if the specific data instance is considered a joint of several point anomalies. Therefore, defining the boundary value of the anomalies is crucial. Figure 2.1a provides an illustrated example of this type.

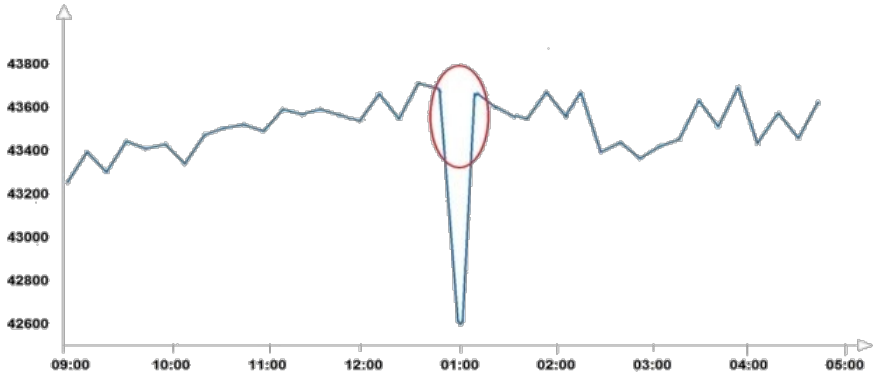
2. **Contextual Anomalies**
   A contextual anomaly or a conditional anomaly is defined based on the data's structure. Often, contextual anomalies are identified by considering both contextual and behavioral features [12]. Depending on the context the problem is defined, a data point can be defined as normal or anomalous based on the exterior behavioral point of view that explicitly describes that context. Time series data is a classical example of defining contextual anomalies. For example: observing data over time series or time-related concepts such as days, weeks, months, and time off can disclose anomalous instances directly related to such anomalies.

3. **Collective Anomalies**

   Collective or group anomalies refer to a collection of related data points or instances that are anomalous concerning the entire dataset. An individual data point in a group anomaly may not be anomalous within themselves but may occur together as a collection or anomaly group. In time series data, a collective anomaly can be identified as a peaking value in a particular time and stays in the medium range throughout the rest of the series.



(A) Point Anomaly (B) Contextual Anomaly



(C) Collective Anomaly

FIGURE 2.1: The subplots illustrate the three types, as mentioned earlier, of anomalies [1]. 2.1a presents a point anomaly. Data points marked as red are anomalous since they do not belong to the standard data points group (green). 2.1b represents a group anomaly (green points) since those group of instances derives further from the standard behavior. Finally, 2.1c forms a contextual anomaly compared to their normal series of behaviors.

2.2.2 Data Labels

The labels attached to a data point denote if the data point is normal or anomalous. Data labeling is often done manually by humans; therefore, it requires substantial

effort to acquire the correct label [11]. Likewise, anomalous behavior can change if the data instance attribute changes. For instance, data points with seemingly normal collective behavior are more difficult to detect within distribution-based group instances. In these cases, point wise anomaly detection methods can be used in group anomalies by characterizing certain properties. Compared to the expected group pattern in image-based applications, distribution-based anomaly detection methods have a distinctive combination of visual features.

Given a dataset $\mathcal{D} = \left\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\right\}$, two possible scenarios can happen for AD methods:

- **Supervised**: We can assume the normal and anomalous or outlier data are labeled accordingly.
- **Unsupervised**: Both standard and anomalous data can be unlabeled and mixed.

Based on the context in which the labels are available, the AD models are roughly categorized into the following three classes:

1. **Supervised Anomaly Detection (SAD)**
   The supervised anomaly detection models train a supervised multi-class or binary classifier using both normal and anomalous data class labels. In general, the SAD techniques are classified into Decision Tree (DT), Support Vector Machine (SVM), and Neural Network (NN) based methods [11]. In the majority, the performance of a supervised classifier uses a sub-optimal class imbalance to detect an outlier, resulting in better accuracy than semi-supervised and unsupervised methods [13]. Although the supervised models' accuracy is superior, there is a substantial chance of misclassifying normal and anomalous classes in complex and non-linear feature space. Multi-Class supervised models are a potential solution to this complexity [14]. Multi-Class classifier learns to distinguish anomalous classes from all data instances. Generally, the Multi-Class models are divided into two sub-networks: a feature extractor network followed by a sub-classifier [15]. Max-Margin Classifier [16], Effective Discrimination [17], and SVM [18] are some of the most compelling feature classifiers to date. However, while training with high-dimensional

data that learn the image's hidden representation, the classifier's complexity is a concern. The training complexity increases linearly with the number of hidden layers and requires considerable computational complexity to learn those models.

2. **Semi-Supervised Anomaly Detection (SSAD)**
   Unlike the SAD methods, in SSAD techniques, the training set has only labeled instances of the normal class and does not require labels for the anomaly class. Compared to SAD and UAD, SSAD performs well under the assumption that only data from a single class is available for training. As a consequence, the multi-class classification does not achieve satisfactory performance on class imbalance problems. Local Outlier Factor (LOF) [19, 20], One-Class Support Vector Machine (ocSVM), iForest [21, 22], Support vector data description (SVDD) [20, 23, 24], and Naïve Bayes [25, 26] are a few SSAD methods that efficiently work with such one-class classifications. However, due to the lack of objects representing normal classes, the SSAD often suffers from outlining the boundaries between normal classes and anomalous classes. Min et al. [27] proposed such methods to learn the discriminative boundary throughout the normal data points. Additionally, training autoencoders in conjunction with a one-class method is a widely exercised process of SSAD [28]. Techniques like Corrupted Generative Adversarial Networks (CorGAN) [29], RandNet autoencoder, Deep Feature Consistent VAE [30], and EncoderForest(eForest) [31] presented similar ideas of learning hidden representation among the high-dimensional image dataset by extracting attributes from highly complex feature space. For such training techniques, losing information and reconstruction are often observed. Adequate training data of normal class would produce low constructions error for autoencoders over anomalous instances [32, 33]

3. **Unsupervised Anomaly Detection (UAD)**
   Methods that function in unsupervised techniques implicitly assume that the anomalous data is distant from the normal data. As a result, the unsupervised methods do not require training data labels, making these techniques widely popular and adopted. Recurrent Neural Networks (RNN), One-Class Support Vector machines, K-means, and Genetic Algorithms are the distinct categories of UAD methods [11]. However, the two most used methods in unsupervised learning are neural network-based models such as Autoencoders

and Transformers. The autoencoders are the typical architecture where the optimization problem is generally non-convex and has a quadratic computational cost. The computational cost depends on the parameter, hidden layers, and the total operation performed. Compared to Principal Component Analysis (PCA) and Spatial Transformer Networks (STN) [34]; Adversarial Autoencoders (AAE) [35], Variational Autoencoders (VAE) [36] and, Denoising Autoencoders (DAE) [37] have higher computational complexity since they are not based on matrix decomposition. Furthermore, Semi-Supervised methods can also be transformed into unsupervised methods simply by removing the data label of anomalous class. Additionally, Meng et al. [38] proposed relational encoders for feature extraction for better interpretation; however, such UAD methods lack accuracy due to the higher sensitivity of noise and corruption data.

2.2.3 Output of Anomaly Detection

Although the training method for any anomaly detection model is crucial, the model's output or reporting aspects also play an important role. The output value enables the results interpretation process and helps to understand the workflow. Typically, it is categorized into two types:

1. **Anomaly Score**
   Based on the degree of anomaly and domain-specific threshold, the anomaly score technique attaches a score to each data point. In general, this score defines the level of "*outlierness*" for each data instance [11]. Sometimes, this scoring approach is a decision-making criterion to determine the distance or how far the actual data points are from average scores.

2. **Anomaly Label**
   Although the anomaly score is an effective way to rank the data instance, the anomaly label helps to analyze domain-specific threshold to identify the most relevant outliers. The binary labels allow controlling of the parameter or magnitude of the reconstruction errors within each technique.

Classifying anomalous and normal classes from unlabeled data is more challenging and is the primary domain this thesis focuses on. Specifically, identifying anomalous data points from image is a more challenging task as the image dataset contains a more complex hidden representation and higher dimensions of factors.

2.3 UAD Techniques

As discussed in Chapter 2.2.2, based on the label of the data, the outlier detection methods are roughly categorized into supervised, semi-supervised, and unsupervised learning. Considering the scope of this thesis, we confined our next step of literature to unsupervised methods. In unsupervised learning, it is widely assumed regarding anomaly detection that learning the density distribution is more complex than learning the boundary of data. UAD techniques are categorized into clustering and manifold learning methods based on data mass boundary and distribution. Principal component analysis (PCA) and autoencoders are two other general categories where the linearity of the data is considered as the separation criteria. Table 2.1 provides a generic classification of UAD techniques discussed in this thesis.

Table 2.1: UAD Techniques

Learning Type	Techniques
Clustering	Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [39]
	Expectation-Maximization (EM) [40]
	K-Means Clustering [41]
Manifold Learning	Isomap Embedding [42]
	Locally Linear Embeddings (LLE) [43]
	Hessian Eigenmapping [44]
	Multi-dimensional Scaling (MDS) [45]
	t-distributed Stochastic Neighbor Embedding (t-SNE) [46]
PCA	Kernel-PCA [47]
	Robust PCA [48]
	Logistic PCA [49]
	Quadratically regularized PCA [50]

	Singular value decomposition (SVD) [51]
Autoencoder (AE)	Non-generative Autoencoders
	Variational Autoencoders
	Regularized Autoencoders

Data visualization and observing the hidden pattern of the data get significantly more complex for a high-dimensional dataset. More often, the number of samples in the dataset is much lower than the number of dimensions in each data sample. To aid the visualization of data, learn latent representation, and disentanglement learning, effective application of AD methods is required to retain the structure, and essential features of the data as well as not lose valuable information. PCA uses a linear technique to capture maximum data variance by finding low-dimensional projections. Although PCA solves the problem of complex data visualization, it often fails to capture the mapping of non-linear data. Another variation of PCA, Kernel-PCA or KPCA, introduces the idea of non-linearity in PCA to overcome complex dimension problems and achieve implicit mapping. However, PCA and KPCA need to be regularized as they become inconsistent estimators due to fewer replicates/samples than measure features [52].

Similar to KPCA, Manifold learning generalizes the linear structure to learn the smooth manifold curve from the multi-dimensional space. Compared to PCA and KPCA, the manifold learning algorithms such as t-SNE, MDS, and LLE perform better in capturing the non-linear relation of the complex data. However, when we consider the task of unsupervised anomaly detection, the embedding space of the data often gets non-linear, and the latent representation transforms into a complex projection. The effectiveness of manifold learning drops back in such cases [53].

2.3.1 Why Autoencoders?

In general, the task of dimension reduction and AD in an unsupervised manner is divided into the following sub-problem:

1. Visualize the high-dimensional representation of the data into a low-dimensional subspace (i.e. compressing and compacting the original data).

2. Learn the latent variable representation from the low-dimensional space.

3. Using that learning outcome from latent representation, reconstruct the original or standard data points, and categorize the anomalous data points using the reconstruction error.

The advantage of autoencoder over other UAD techniques is the powerful ability to learn the complex latent representation and reconstruct original input. For instance, recently proposed methods show that the PCA, KPCA, and their variants, including Sparse PCA and Robust PCA, can only reduce the dimension for linear and non-complex hyperspace data [52]. The latent representation for such models cannot capture complex patterns since it only performs better with linear data [53]. Hence, KPCA methods and manifold learning, such as t-SNE and LLE, lacks to reconstruct multi-dimensional complex hyperspace data. In contrast, the autoencoder combines the recognition modeling (encoder) and generative regularization modeling (decoder) techniques to learn compressed representation and construct the latent representation of the data simultaneously. Additionally, the autoencoder models the low dimensional latent space without losing substantial information from compressed input or code, mostly known as tackling the curse of dimensionality of the complex data. Motivated by such advantages, this thesis focuses on the autoencoder and its extensions.

2.4 Taxonomy of Autoencoders

Autoencoders are deep neural networks trained to reproduce their input as their output. AE learns the hidden representation of the input, h and reconstructs the original input as output. In general, the learning process is defined as an encoder $f(x)$, and the final output defined as a decoder, $g(f(x))$, $\tilde{x} = g(h)$. In some architectures, the latent representation can be defined as the bottleneck that holds the compressed information from the input code h.

2.4.1 Variational & Generative Modeling

Recently, VAE and Generative Adversarial Networks (GANs) stood up as the two most powerful approaches to generative modeling. VAE enhances classical autoencoders by including a Bayesian component to learn structure representing the probability distribution of the data and imposing a prior on the probability of the encoder [54]. VAE models the Gaussian random variable that results in the regularization to explain the probability of the input implicitly. Moreover, VAE avoids the marginal likelihood probability estimation by introducing a variational lower bound. Another complication of estimating the Markov chain sampling process is resolved by the reparameterization trick. Combing such techniques enables controlling the distribution of the latent representation vector $\mathbf{z}$ and thus contribute to effective sample generation. Concise technical details and the mathematical foundation is mentioned in Chapter 3.3.5.

Despite having the aforementioned advantages, VAE suffers from blurry image generation, limitations in priori to track mixed data distributions, less interpretable latent representation, and, most prominently suffering, the "curse of dimensionality" [55]. To solve the limitations mentioned above, modifications to the VAE architecture were proposed based on several criteria: regularization of the encoding distribution [5], learning disentangled and discrete representations [2, 56, 57], Prior Variant [58], regularizing posterior-variant [4, 59], conditional generations [60], extracting continuous features [37, 61], optimizing Gaussian mixture [62], modeling distributions on discrete variables [63], learning correlated features [64], data clustering, and generation [65]. Such extensions have shown great promise to learn the data distribution and generate meaningful generations from encoded distribution. However, using current VAE architectures, addressing both learning disentangled factors and improving anomaly detection accuracy still remain a challenging research domain.

Unlike VAE, GAN attempts to balance the generator and the discriminator to produce more empirically higher quality and higher definition results than variational models. However, GANs are criticized for mode collapse (i.e. the generator only generates data from a single mode of the distribution) and inflexibility in evaluating performance on UAD tasks. Moreover, GAN lacks in sample generations in a context where the negative training instances exceed positive instances. DCGAN

[66], CGAN [67], Info GAN [68], and WGAN [69] are a few GAN architectures to address these issues.

2.4.2 Learning Meta-Priors

Bengio et al. [70] introduced the meta-priors for unsupervised representation learning tasks. According to Bengio et al. [70], "the meta-priors are derived from general assumptions about the world, such as the hierarchical organization or disentanglement of explanatory factors, the possibility of semi-supervised learning, the concentration of data on low-dimensional manifolds, clusterability, and temporal and spatial coherence [71]." Considering the scope of this thesis, learning disentanglement representation and regularizing total correlation are two preliminary meta-priors focused on in this section.

Learning disentangled factors is critical for efficient downstream tasks and improving reconstruction quality. Assuming the data is initialized with variations of independent variables in the representation, disentangled factors often depend on multiple variables. As an accord, similar factors should control distinct features, and learning those features results in structured representation learning. For example, shapes, colors, and strokes in the MNIST [72] dataset are controlled by distinct independent factors. VAE architectures concentrated on regularizing posterior-variant, such as β-VAE and its variants [2, 3, 73], FactorVAE [4], Relevance FactorVAE (RFVAE) [59], InfoVAE [5], and VFAE [6], mainly focusing on this representation learning issue.

On the other hand, The principle of Total Correlation Explanation (CorEx) introduces the learning process of disentangled factors and interpretable representations utilizing multivariate information theory [74]. Instead of assuming the data is generated using independent factors, CorEx inspects latent variable $\mathbf{z}$ to retrieve meta-priors and provides insights into progressive disentangled learning. Previously, Ver Steeg and Galstyan [75], Gao et al. [74], Ver Steeg and Galstyan [76], Ver Steeg and Galstyan [77] proposed methods based on multivariate information theory to learn featured representations. Additionally, Chen et al. [78], Khemakhem et al. [79], Kim et al. [80] emphasizes regularizing total correlation (TC) or total correlation loss (TC-loss) in an information-theoretic way.

2.5 Connecting VAE Architectures

In this thesis, we propose a new architecture that captures the disentangled representation of the data using the extension of CVAE and CorEX as well as modeling latent variables and data, both conditioned on known sources of variation.

β-VAE, CVAE, and CorEx heavily depend on maximizing the latent representation's informativeness. With VAEs, we introduce a recognition model $q_\phi(\mathbf{z} \mid \mathbf{x})$ whose purpose is to approximate $p_\theta(\mathbf{z} \mid \mathbf{x})$. The goal is then to derive a variational bound that can be optimized. Consequently, the recognition model (i.e., the encoder) can be learned jointly with the generative model (i.e., the decoder). Two interesting observations are derived from the recognition model of VAE:

- What are the generative factors of the data that are captured by the latent variable? The latent variable is usually heavily diverse since the factors are encoded in the multiple interdependent components of $\mathbf{z}$.
- How can we overcome the limitation of generating data with VAE, which does not have any control over the kind of data it generates? In general, the VAE model can be trained to generate random samples with good reconstruction quality; however, it has no control over generating specific samples.

Current literature shows β-VAE attempts to force $\mathbf{z}$ to learn a disentangled representation of the data (e.g., to force the components of $\mathbf{z}$ to be independent). β changes the degree of applied learning pressure during training that encourages different learned representations. Compared to VAE, where $\beta = 1$, enforcing stronger constraint $\beta \geq 1$ on the latent bottleneck encourages the model to learn the most efficient or disentangled representation by conditioning independent factors. However, while constraining the value of β, a trade-off between reconstruction fidelity and the quality of disentanglement is often observed. As a consequence, penalizing $KL(q(z)\|p(z))$ makes $q(z)$ to factorized as prior $p(z)$ in most scenarios. On the other hand, penalizing $I(x; z)$ reduces the amount of information about x stored in z, which leads to poor reconstruction. Therefore, current literature still shows limitations in addressing the first observation.

For the second observation, the current architectures of CVAE provide an excellent graphical modeling approach. CVAE is necessarily another extension of VAE

where conditioning on known sources c is added directly to the encoding and decoding processes. In terms of conditional distributions, we want to investigate $p_\theta(\mathbf{x} \mid c)$ instead of $p_\theta(x)$. In other words, the generative model of CVAE becomes $p_\theta(\mathbf{x} \mid \mathbf{z}, c)$, and the recognition model becomes $q_\phi(\mathbf{z} \mid \mathbf{x}, c)$. On the other hand, CorEx is the opposite of VAE and CVAE in terms of choice of prior. The objective function of CorEx defines an encoder $p_\theta(\mathbf{z} \mid \mathbf{x})$ and evolves a decoder $q_\phi(\mathbf{x} \mid \mathbf{z})$ via variational approximation to the true posterior [74]. However, instead of the essential variational approximation to the posterior in VAE, CorEx requires variational distribution as true data distribution $p_\theta(\mathrm{x})$, which is unknown or intractable. Consequently, utilizing conditioning variables with information theory can greatly benefit learning disentangled representation by controlling sample approximation.

Considering the observations mentioned above, the subsequent two manuscripts seek to explain the potential anchoring ground for VAE architectures and connect the UAD frameworks. Firstly, Chapter 3 explores VAE and its extensions to analyze several UAD downstream tasks described above. Then, Chapter 4 proposes dCVAE to address the two questions above.

Chapter 3

A Comprehensive Study of Autoencoders for Anomaly Detection: Efficiency and Trade-Offs

Preamble to Manuscript 1. As I discussed in Chapters 1 and 2, VAE architectures are diversely utilized in miscellaneous applications. Therefore, finding the exact mathematical foundation and observing the changes in both qualitative and quantitative approaches are crucial. Over the summer of 2021, Dr. Turgeon and I started going over the mathematical foundation of different VAEs. We discussed the formulation of VAEs over other methods, learning how objective function works, decoding the operations of generative models, and gathering resources for reproducing the current autoencoder architectures. In general, the purpose of the manuscript presented in this chapter was to review various autoencoder methods and rank them based on their overall performance in AD tasks. Additionally, we seek to observe the following:

- Sensitivity of autoencoder in a noisy dataset and how they respond to such environment.
- An efficient way to minimize cost function weights for re-sampling in the Evidence Lower Bound (ELBO)

- Observe how the autoencoders retain information in lower dimensions while optimizing the reconstruction quality.
- Perceiving the trade-offs among different architectures.

Through the study of performance and limitations of these architectures, I identified the important properties that an autoencoder architecture must possess in order to provide good performance on AD tasks.

This manuscript will soon be submitted at the IEEE Transactions on Knowledge and Data Engineering (TKDE) journal.

A Comprehensive Study of Autoencoders for Anomaly Detection: Efficiency and Trade-Offs

Asif Ahmed Neloy
Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada

Maxime Turgeon
Department of Statistics
University of Manitoba
Winnipeg, MB, Canada

Abstract

Unsupervised Anomaly detection (UAD) is a diverse research area with application in many different domains. Over the years, many anomaly detection techniques such as clustering, generative modeling, or variational inference-based methods have been developed to solve certain deficiencies and improve on state-of-the-art techniques. Recently, deep learning and generative models (such as autoencoders) have led to major advances in the field of UAD. In this study, we review 11 autoencoder (AE) architectures divided into three distinct categories and conduct experiments to understand their capabilities in four different domains: data reconstruction, sample generation, latent space visualization, and anomaly detection. Throughout the experiments, we also carefully monitor the scope of reproducibility with different training parameters. Fashion-MNIST (FMNIST) and MNIST are the two selected datasets utilized in these experiments. Finally, we discuss the advantages, trade-offs, and limitations of different AE architectures for future research directions.

3.1 Introduction

Anomaly detection (AD), also referred to as Outlier Detection (OD) or Novelty Detection (ND), is an application of machine learning that focuses on detecting unconventional observations in a sample dataset [11]. Primarily, AD focuses on identifying noisy or erroneous measurements in a dataset based on the lower similarity compared to standard observations. In other words, an anomaly is a data point likely to result from a different data distribution than normal data [81].

By their very nature, anomalies are rare events. Often, the definition of anomalies heavily depends on the set of a problem or the context of the data. In close problem contexts, where the boundaries between normal and anomalous data are not clearly drawn, traditional deep learning models, including supervised and hybrid learning approaches to anomaly detection, tend to perform poorly due to a significant imbalance between anomalous and normal classes. Similarly, accurate labeling of anomalies can be expensive and time-consuming. Therefore, most training datasets for AD have either no labels or very few labels. Unsupervised and semi-supervised approaches have thus shown more success in AD tasks than supervised approaches [12]. For example, unsupervised and semi-supervised approaches can potentially find unknown types of anomalies, whereas supervised approaches can only learn the labeled anomaly types.

In the era of big data, data is increasingly more complex and consists of higher dimensions. Therefore, there is a real need for unsupervised learning methods that deal with unlabeled and complex high-dimensional data. In recent years, deep learning architectures have successfully produced effective data representations using supervised and semi-supervised methods in AD tasks [82]. Specifically, recent work on Variational Autoencoders (VAEs) [54] and Generative Adversarial Networks (GANs) [83] has shown great promise. Both approaches are *generative*: they estimate a latent representation of the data, along with the decoding function that can generate the data starting from the latent distribution.

Generative models are especially well suited for AD. Indeed, by comparing a data point with its reconstruction, we can naturally define an anomaly score. A good reconstruction can be interpreted as evidence that the observation comes from the main distribution, and therefore it is unlikely to be an anomaly. The presence of

some labeled data can then be used to tune cutoff values for the reconstruction error.

From an accuracy perspective, GANs are capable of generating highly realistic images. However, from a computational perspective, GANs are known to be difficult to tune accurately, and they suffer from the "mode collapse" phenomenon (i.e., data is generated around a single mode of the distribution) [69]. On the other hand, even though VAEs may be less accurate than GANs for some tasks, they are easier to train and do not suffer from mode collapse.

The most significant advantage of VAEs over other unsupervised techniques is their powerful ability to learn complex latent representations and reconstruct original inputs using the generative modeling background. VAE combines the method of generative modeling and regularization to compress the original data and then reconstruct it from its compressed form simultaneously by controlling the latent distribution.

3.1.1 Focus and organization of this study

In this article, we review different VAE architectures and assess their suitability for AD using benchmark datasets. For the sake of completeness, we also include non-generative autoencoders. To be specific, this review seeks to address the following objectives:

1. Present a mathematical background for VAE architectures to understand the generative functions better.
2. Conduct a comparative analysis of reconstruction quality, sample generation, and latent space for each AE.
3. Regulate downstream tasks of AD and draw conclusions about *efficiency vs. trade-offs* for future research avenues.

The remainder of this paper is organized as follows: Section 3.2 provides further background information on anomaly detection and autoencoders. Then Section 3.3 presents the different architectures included in our study. Section 3.4 describes the different model configurations and our experimental setup. We use the MNIST

and FMNIST datasets to conduct our experiments and follow the parameters and hyper-parameters mentioned in the main reference research papers (wherever possible). Finally, in Section 3.5, the experimental results are presented and discussed. The paper concludes with Section 3.6 presenting the future research directions for VAEs in AD tasks.

3.2 Taxonomy of Autoencoders

AEs are generally used in various areas of application, including semi-supervised learning, generative modeling, clustering, dimension reduction, unsupervised anomaly detection, data compression, and information retrieval. Considering the different architectures, regularizing methods, and encoding processes, the AEs are generally categorized into the following three categories:

1. **Non-Generative AEs**: Non-Generative AEs are originally proposed as artificial neural networks that perform dimension reduction [84]. Similar to non-linear PCA, the idea of AE is to build input and output layers retaining identical dimensions. Additionally, constructing an intermediate layer (referred to as a bottleneck) to reduce the dimension and project the data into lower dimensions. Undercomplete AE is a classic example of a non-Generative single layer autoencoder that uses a linear activation function in combination with the mean squared error loss function, which results in an equivalence of the PCA algorithm [85]. Denoising autoencoder (DAE) [86], stacked convolutional autoencoder (scAE) [87], and robust deep autoencoder (RDA) [88] are a few extensions of AEs modifying the encoding and decoding functions.

2. **Variational Models**: Variational Autoencoders (VAE) are a special type of AE that extends the original AE architecture by using the variational Bayes inference and generative modeling frameworks. The main difference between AE and VAE is that VAE includes a Bayesian model that learns the compressed representation of data and generates samples using that input data. The primary goal of VAE is to learn the representation of the data and generate samples using that learned representation. In general, VAE is a combination of a recognition model or encoder followed by a generative

model or decoder. Using the same component and architecture of VAE, several extensions have been proposed. In this review, we chose the following extensions of VAE: Self-Adversarial Variational Autoencoder (adVAE) [89], β-VAE [2], Conditional Variational Autoencoder (CVAE) [60], and Vector Quantized Variational Autoencoder (VQ-VAE) [58].

3. **Regularized Models**: Regularized AEs are the extension of VAE and AE architectures that combine the following objectives: latent regularization, learning disentangled and discrete representations, and regularizing posterior-variants. Often, regularized models combine "bottleneck" techniques with a probabilistic perspective to enforce restrictions into the VAE architecture (e.g., introducing sparsity, weighted loss functions, and probabilistic modeling). Considering the scope of this review, we chose: Adversarial Autoencoder (AAE) [90], Sparse Autoencoder (SAE) [91], Contractive Autoencoder (CAE) [92], Importance Weighted Autoencoder (IWAE) [93], and Probabilistic Autoencoder (PAE) [94] from this category.

The AE architectures selected from the three criteria mentioned above, we considered the elemental architectures introduced from each criterion. For example, we chose VAE as the first variational model that introduced the Bayesian modeling approach into an AE architecture, and β-VAE included constraints to control disentangled factors. The authors intended to investigate the architecture in a fundamental structure to propose further scopes.

3.3 Mathematical Preliminaries

This section gives the mathematical formulation of each autoencoder architecture that appears in the experiments in Section 3.4. Table 3.1 summarizes the notation used throughout the paper.

TABLE 3.1: Main notation used throughout the paper

Notation	Definition
$\mathcal{D}$	The training dataset, $\mathcal{D} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\}$, with n data points.

$\mathbf{X}^{(i)}$	Each data point i is a vector of d dimensions, $\mathbf{x}^{(i)} = \left\{x_1^{(i)}, x_2^{(i)}, \ldots, x_d^{(i)}\right\}$
$\mathbf{X}$	Sample data point from Dataset, $\mathbf{x} \in \mathcal{D}$
$\mathbf{x}'$	Reconstructed data from input data $\mathbf{X}$
$\tilde{\mathbf{x}}$	Corrupted data point
$\mathbf{Z}$	Compressed data from bottleneck layer
$\mathbf{W}$	$d' \times d$ weight matrix
$\mathbf{b}$	Bias vector
$a_j^{(l)}$	Activation function with j-th neuron and i-th hidden layer
$g_\phi(.)$	Encoder function with ϕ parameter
$f_\theta(.)$	Decoder function with θ parameter
$q_\phi(\mathbf{z} \mid \mathbf{x})$	Probabilistic encoder or estimated posterior probability function
$p_\theta(\mathbf{x} \mid \mathbf{z})$	Probabilistic decoder or likelihood function.

3.3.1 Basic Autoencoders

The basic architecture of autoencoder is simply a neural network designed to learn the hidden representation $\mathbf{h} \in \mathcal{R}^{d'}$ of the input data x $\in \mathcal{R}^d$ in an unsupervised way. The hidden representation is often denoted as an identity or deterministic function $\mathbf{h} = f_\theta = \sigma(Wx + b)$, where $\theta = \{W, b\}$, that reconstructs the original input by compressing the input data. The core idea of autoencoders was first discussed in the 1980s, and it was later promoted by Hinton and Salakhutdinov [61]. The basic autoencoder contains an encoder function $g_\phi(.)$ parameterized by ϕ, a bottleneck layer $\mathbf{z} = g_\phi(\mathbf{x})$ with an input of x and reconstructed output $\mathbf{x}' = f_\theta\left(g_\phi(\mathbf{x})\right)$, where the decoder function $f_\theta(.)$ is parameterized by θ.

3.3.2 Denoising Autoencoder (DAE)

The objective of DAE is to reconstruct the original data x using a noisy version $\tilde{\mathbf{x}}$ as input. Noisy reconstruction resolves several limitations of non-generative autoencoder such as over-fitting, learning compressed data representation and generate samples from corrupted input [86]. The input is corrupted by adding noise or masking in the input layer by input vector in a stochastic manner. the objective function of DAE combines $\tilde{\mathbf{x}} \sim \mathcal{M}_{\mathcal{D}}(\tilde{\mathbf{x}} \mid \mathbf{x})$ and defined as:

$$\begin{aligned} \tilde{\mathbf{x}}^{(i)} &\sim \mathcal{M}_{\mathcal{D}}\left(\tilde{\mathbf{x}}^{(i)} \mid \mathbf{x}^{(i)}\right) \\ L_{\mathrm{DAE}}(\theta, \phi) &= \frac{1}{n} \sum_{i=1}^{n} \left(\mathbf{x}^{(i)} - f_{\theta}\left(g_{\phi}\left(\tilde{\mathbf{x}}^{(i)}\right)\right)\right)^2 \end{aligned} \tag{3.1}$$

Here, $\mathcal{M}_{\mathcal{D}}$ refers to the mapping from the original data samples to the noisy ones. In order to remove the noise from original input $\mathbf{x}$ and reform the reconstructed sample as $\mathbf{y} = f_{\theta'}(h) = \sigma\left(W'h + b'\right)$. Finally, the denoising autoencoder is trained to find the latent representation $\mathbf{h} = f_{\theta}(\bar{x}) = \sigma(W\bar{x} + b)$.

3.3.3 Sparse Autoencoder (SAE)

The main motivation of sparse autoencoder (SAE) is to learn the sparse representation using only a single hidden layer with a feed-forward neural network [91]. In SAE, a sparsity enforcer of L1 regularization directs a single-layer network to learn and minimize the error in reproducing the input. A simple construction of SAE is illustrated in Equation 3.2:

$$\widetilde{\mathrm{x}} = H_{W,b}(\mathrm{x}) \approx x. \tag{3.2}$$

Here, x denotes the input vector, and $\widetilde{\mathrm{x}}$ is the output vector. The weight matrix and this part's bias are represented by $W^{(1)}$ and $b^{(1)}$, respectively. The decoder connects the corresponding weight matrix $W^{(2)}$ and $b^{(2)}$ with the hidden layer to the output layer. Compared to the basic architecture of autoencoder, the SAE introduces *sparsity constraint* by enforcing H. Sparsity constraint controls the weight and bias to have a small number of layers activated at the same time. Finally, using

the same weight W and bias b parameters, $H_{W,b}(\mathrm{x})$ acts as a non-linear function of SAE that predicts the output vector $\widetilde{\mathrm{x}}$ based on the input vector x.

3.3.4 Contractive Autoencoder (CAE)

Similar to SAE, the contractive autoencoder (CAE), proposed by Rifai et al. [92] aims to learn a sparse latent representation, but it is more robust to small variations in the data. CAE is another regularization technique, just like DAE and SAE. However, in order to penalize the sensitivity to input representation, CAE adds a variable term in the loss function. With this extra term, CAE improves the robustness to small variations. The Frobenius norm of the Jacobian matrix $J_f(x)$ of the encoder measures the sensitivity with respect to input and can be represented as follows:

$$\left\|J_f(\mathbf{x})\right\|_F^2 = \sum_{i,j}\left(\frac{\partial f_j(\mathbf{x})}{\partial x_i}\right)^2. \tag{3.3}$$

Using Jacobian matrix from Equation 3.3, the loss function of CAE is defined as:

$$\mathcal{J}_{\mathrm{CAE}}(\theta) = \sum_{x\in D_n}\left(L(x, g(f(x))) + \lambda\left\|J_f(x)\right\|_F^2\right). \tag{3.4}$$

The idea of Equation 3.4 is to penalize the $\left\|J_f\right\|_F^2$ term while staying more invariant such that it carves a representation that corresponds to a lower-dimensional non-linear manifold.

3.3.5 Variational Autoencoder (VAE)

The concept of Variational Autoencoder (VAE, proposed by Kingma and Welling [54]) adapts the idea of variational inference for graphical models into the autoencoder framework. Instead of mapping the input into a fixed vector, the VAE maps into a distribution which results in a specific estimator for a training algorithm called Stochastic Gradient Variational Bayes (SGVB). For a distribution denoted by p_θ and parameterized by θ, the association between the input $\mathbf{x}$ and the latent

encoding vector $\mathbf{z}$ is defined by a prior $p_\theta(\mathbf{z})$, a likelihood $p_\theta(\mathbf{x}|\mathbf{z})$, and a posterior $p_\theta(\mathbf{z}|\mathbf{x})$. Since the posterior is typically intractable, we approximate it using a distribution $q_\phi(\mathbf{z}|\mathbf{x})$.

In order to quantify the distance between the posterior and its approximation, Kingma and Welling [54] use the Kullback-Leibler divergence (KLD) $D_{\mathrm{KL}}(q \mid p)$. The main idea is to minimize $D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \mid p_\theta(\mathbf{z}|\mathbf{x}))$ with respect to ϕ using the reversed KL $D_{\mathrm{KL}}(Q|P) = \mathbb{E}_{z\sim Q(z)} \log \frac{Q(z)}{P(z)}$ and Forward KL divergence $D_{\mathrm{KL}}(P|Q) = \mathbb{E}_{z\sim P(z)} \log \frac{P(z)}{Q(z)}$. Furthermore, using conditional distribution $p_{\theta^*}(\mathbf{x}|\mathbf{z} = \mathbf{z}^{(i)})$, VAE reconstructs the encoding vector as $p_\theta\left(\mathbf{x}^{(i)}\right) = \int p_\theta\left(\mathbf{x}^{(i)} \mid \mathbf{z}\right) p_\theta(\mathbf{z}) d\mathbf{z}$. Additionally, to derive the ELBO bounds, we imposes the KLD and update the approximation as:

$$\begin{aligned} D_{\mathrm{KL}}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p_\theta(\mathbf{z} \mid \mathbf{x})\right) = \log p_\theta(\mathbf{x}) + \\ D_{\mathrm{KL}}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p_\theta(\mathbf{z})\right) - \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x} \mid \mathbf{z}) \end{aligned} \tag{3.5}$$

Based on Equation 3.5, we can define the loos function $L_{\mathrm{VAE}}(\theta, \phi)$ as follows:

$$\begin{aligned} L_{\mathrm{VAE}}(\theta, \phi) &= -\log p_\theta(\mathbf{x}) + D_{\mathrm{KL}}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p_\theta(\mathbf{z} \mid \mathbf{x})\right) \\ &= -\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x} \mid \mathbf{z}) + D_{\mathrm{KL}}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p_\theta(\mathbf{z})\right). \end{aligned} \tag{3.6}$$

Finally, in Variational Bayesian Models, the loss function is often referred to as the *variational lower bound* or *evidence lower bound*. As the KLD is always non-negative, $-L_{\mathrm{VAE}}$ is a lower bound for $\log p_\theta(\mathbf{x})$. Therefore, minimizing the loss function is equivalent to maximizing the evidence lower bound:

$$\begin{aligned} -L_{\mathrm{VAE}} &= \log p_\theta(\mathbf{x}) - D_{\mathrm{KL}}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p_\theta(\mathbf{z} \mid \mathbf{x})\right) \\ &\leq \log p_\theta(\mathbf{x}). \end{aligned} \tag{3.7}$$

The loss function 3.6 of VAE consists of two terms: the first maximizes the reconstruction likelihood by penalizing reconstruction error; and the second term encourages learning distribution, similar to the true prior distribution.

3.3.6 Conditional Variational Autoencoder (CVAE)

The Conditional Variational Autoencoder (CVAE), proposed by Pol et al. [60], is an extension of VAE where the generative model is conditioned on known factors $\mathbf{c}$. Compared to VAE, CVAE generates data by modelling the latent variable $\mathbf{z}$ using both the input $\mathbf{x}$ and the class information of each instance. Recall Equation 3.7 from VAE:

$$L_{\mathrm{VAE}}(\theta, \phi) = -\mathbb{E}_{\mathrm{z}\sim q_\phi(\mathrm{z}|\mathrm{x})} \log p_\theta(\mathrm{x} \mid \mathbf{z}) + D_{\mathrm{KL}}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p_\theta(\mathbf{z})\right). \tag{3.8}$$

CVAE achieves this goal by learning a decoder $p_\theta(\mathbf{x} \mid \mathbf{z}, \mathbf{c})$ and a recognition model $q_\phi(\mathbf{z} \mid \mathbf{x}, \mathbf{c})$, where $\mathbf{c}$ is the conditioned variable. Updating the VAE function with this conditioned variable, we get:

$$L_{\mathrm{CVAE}}(\theta, \phi) = -\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{c},\mathbf{x})} \log p_\theta(\mathbf{x} \mid \mathbf{z}, \mathbf{c}) + D_{\mathrm{KL}}\left(q_\phi(\mathbf{z} \mid \mathbf{x}, \mathbf{c}) \| p_\theta(\mathbf{z} \mid \mathbf{c})\right). \tag{3.9}$$

Conditioning on $\mathbf{c}$ can be very advantageous for AD tasks, as the extra information can help distinguish between point and contextual anomalies [11]. The variational lower bound (Equation 3.9) for CVAE is derived similarly as for VAE.

3.3.7 β-VAE

β-VAE (proposed by Higgins et al. [2]) is a modification of the VAE objective function that emphasizes learning disentangled latent factors. Under a small constant δ while keeping the distance between the actual and estimated posterior distributions small, β-VAE maximizes the probability of generating accurate data:

$$\begin{aligned} &\max_{\phi,\theta} \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x} \mid \mathbf{z})\right], \\ &\quad \text{where } D_{\mathrm{KL}}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p_\theta(\mathbf{z})\right) < \delta. \end{aligned} \tag{3.10}$$

The representation can be identified as disentangled or factorized when each variable in latent representation is invariant primarily to other factors and only sensitive to one single generative factor [73]. Therefore, the loss function can be defined as:

$$L_{\text{BETA}}(\phi, \beta) = -\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x} \mid \mathbf{z}) + \beta D_{\text{KL}}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p_\theta(\mathbf{z})\right). \tag{3.11}$$

When $\beta = 1$, it represents the original VAE. When $\beta > 1$, the β-VAE limits the representation capacity of $\mathbf{z}$ by applying a constraint on the latent bottleneck. Larger β suggests more effective latent encoding and further encourages disentanglement. As the value of β largely depends on the KLD, it may result in lower latent channel capacity than the number of generating factors. The lower latent channel will result in a lower-rank projection of the true data generative factors.

3.3.8 Self-Adversarial Variational Autoencoder (adVAE)

Traditional VAEs only regularize the encoder network and not the decoder network. As a consequence, it is difficult to create powerful and expressive decoders from such architecture, as they tend to overfit the data and learn a representation that is not dependent of the latent variables. As we will illustrate later, this is especially problematic for unsupervised anomaly detection.

In order to regularize the decoder so that it learns representations that are useful for anomaly detection, adVAE [90] introduces a third network T, called a *Gaussian transformer*. The network T transforms the latent representation of a normal observation into an anomalous latent representation. In a first training step, the decoder D learns to generate normal and anomalous samples that are far apart, while the network T competes with D by trying to reduce the KL divergence between the normal and anomalous codes; during this step, the weights of E are fixed. Then, in a second training step, the weights of D and T are fixed. To update the weights of E, the loss function is designed as a combination of the usual VAE loss function and discrimination factors that explicitly turn E into a discriminator: it directly learns to distinguish between normal and anomalous observations. In this

way, adVAEs combine the generative power of GANs, while completely avoiding the problem of mode collapsing [68].

3.3.9 Importance Weighted Autoencoder (IWAE)

The importance weighted autoencoder (IWAE) uses a similar generative network and recognition network architecture as the VAE [93]. The main difference between VAE and IWAE is the simple derivation of importance sampling in the evidence Lower Bound (ELBO). The loss function of IWAE is defined as follows [93]:

$$\begin{aligned} \mathcal{L}_k(\mathbf{x}) &= \mathbb{E}_{\mathbf{h}_1,\ldots,\mathbf{h}_k \sim q(\mathbf{h}|\mathbf{x})} \left[\log \frac{1}{k} \sum_{i=1}^{k} \frac{p\left(\mathbf{x}, \mathbf{h}_i\right)}{q\left(\mathbf{h}_i \mid \mathbf{x}\right)} \right] \\ &= \mathbb{E}_{\mathbf{h}_1,\ldots,\mathbf{h}_k \sim q(\mathbf{h}|\mathbf{x})} \left[\log \frac{1}{k} \sum_{i=1}^{k} w_i \right]. \end{aligned} \tag{3.12}$$

The right part of the Equation 3.12, $\left[\log \frac{1}{k} \sum_{i=1}^{k} w_i\right]$ is the unnormalized importance weights. The IWAE is trained to maximize the lower bound on $\log p(\mathrm{x})$. To estimate the log-likelihood, the IWAE utilizes this lower bound corresponding to the k-sample importance weighting. Finally, the importance weighted ELBO optimizes by considering a fixed number of samples from the variational distribution and further uses the expectation with respect to Deterministic Encoder:

$$\begin{aligned} \mathcal{L}_k(\mathbf{x}) = \int q\left(\mathbf{h}_1 \mid \mathbf{x}\right) \cdots \int q\left(\mathbf{h}_k \mid \mathbf{x}\right) \\ \left[\log \frac{1}{k} \sum_{i=1}^{k} \frac{p\left(\mathbf{x}, \mathbf{h}_i\right)}{q\left(\mathbf{h}_i \mid \mathbf{x}\right)} \right] d\mathbf{h}_1 \cdots d\mathbf{h}_k. \end{aligned} \tag{3.13}$$

3.3.10 Probabilistic Autoencoder (PAE)

The probabilistic autoencoder (PAE) comprises an AE with a two-stage generative model. After training with a Normalizing Flow (NF), the AE is interpreted probabilistically [94]. As a result, the PAE generalizes the idea of regularization to reduce the effect of undesirable singular latent space variables on non-linear models. Equation 3.14 defines the loss function of PAE containing the Normalizing Flow [94]:

$$\mathcal{L}_{\text{flow-VAE}} = -\mathbb{E}_{p(\boldsymbol{x})}\left[\mathbb{E}_{q_\phi(z|\boldsymbol{x})}\left[\ln p_\theta(\boldsymbol{x} \mid z)\right] - \mathrm{D}_{\mathrm{KL}}\left[q_\phi(z \mid \boldsymbol{x}) \| p_\gamma(z)\right]\right] \tag{3.14}$$

The first part of the objective function of PAE (Equation 3.14) is a likelihood term. The term gives a high value when the encoder and decoder are matched, and the latent variable $\boldsymbol{z}$ makes x_i more closer to the original input. On the other hand, KL-term is a penalizer for normalized latent space probability density estimator. As a result, PAE optimizes the reconstruction ability and the sample quality individually.

3.3.11 Robust Deep Autoencoders (RDA)

The robust deep autoencoder (RDA) is inspired by the Robust Principal Component Analysis (RPCA), which is a method to reduce the sensitivity of PCA to outliers [88]. Equation 3.15 defines the objective function of a RDA:

$$\begin{gathered}\min_{\theta,\tilde{\mathbf{x}}} \left\|x_{input} - f_\theta\left(g_\phi\left(x_d\right)\right)\right\|_2 + \lambda \left\|\tilde{\mathbf{x}}\right\|_{2,1}, \\ \text{where } x_{input} - f_\theta - \tilde{\mathbf{x}} = 0.\end{gathered} \tag{3.15}$$

Similar to RPCA, the RDA splits the input into two parts: $\boldsymbol{L_D}$, d-dimensional input; and $\tilde{\mathbf{x}}$ with λ parameter that tunes the level of sparsity (Equation 3.15). The well-represented data is presented by a hidden layer of encoder $g_\phi(.)$. On the other hand, the noisy and outlier data, which is difficult to reconstruct, is presented by the $\tilde{\mathbf{x}}$. Additionally, as the objective function $\left\|x_{input} - f_\theta\left(g_\phi\left(x_d\right)\right)\right\|_2$ does not specify a particular form of the g and f pair of the Encoder $g_\phi(\cdot)$ and Decoder $f_\theta(\cdot)$ with W weight and b bias for both of them and can be denoted by:

$$\begin{aligned} g_\theta(x) &= g_{W,b}(x) = \text{logit}\left(Wx + b_g\right), \\ f_\theta(x) &= f_{W,b}(x) = \text{logit}\left(W^T g_{W,b}(x) + b_f\right). \end{aligned} \tag{3.16}$$

3.3.12 Vector Quantised-Variational Autoencoder (VQ-VAE)

The vector quantised-variational autoencoder (VQ-VAE) is an extension of VAE that learns discrete latent variables by the encoder. VQ-VAE incorporates the

Vector quantisation method, which uses K-Nearest Neighbour (KNN) to map K-dimensional vectors into a finite set of "code" vectors [58]. After going through the KNN lookup, the encoder's input $E(\mathbf{x}) = \mathbf{z}_e$ becomes the decoder's input $D(.)$:

$$\begin{aligned} \mathbf{z}_q(\mathbf{x}) &= \text{Quantize}(E(\mathbf{x})) = \mathbf{e}_k, \\ \text{where } k &= \arg\min_i \|E(\mathbf{x}) - \mathbf{e}_i\|_2 , \end{aligned} \tag{3.17}$$

$$L = \|\mathbf{x} - D(\mathbf{e}_k)\|_2^2 + \|\text{sg}[E(\mathbf{x})] - \mathbf{e}_k\|_2^2 + \beta \|E(\mathbf{x}) - \text{sg}[\mathbf{e}_k]\|_2^2 \tag{3.18}$$

Here, D is the embedding size, and K is the number of latent variable categories. For the loss function L, $\|\mathbf{x} - D(\mathbf{e}_k)\|_2^2$ denotes the reconstruction loss, $\|\text{sg}[E(\mathbf{x})] - \mathbf{e}_k\|_2^2$ denotes the VQ loss and finally $\beta \|E(\mathbf{x}) - \text{sg}[\mathbf{e}_k]\|_2^2$ denotes the commitment loss. For each part of Equation 3.18, the $\text{sg}[\cdot]$ refers to the stop gradient operator.

3.4 Experimental Setup

The areas of application of UAD are numerous: it can be applied to text, voice, images, videos, and time-series data. However, this study focuses on UAD methods for image data using autoencoders. This type of data is common in applications ranging from biomedical sciences to cybersecurity, genome sequencing, structural fault detection, and disease diagnosis. In this section, we present a description of the experimental setup, datasets, model selection, parameter tuning, and evaluation criteria of the models.

3.4.1 Datasets

Modified National Institute of Standards and Technology (MNIST) [72] and Fashion-MNIST (FMNIST) [95] are the two datasets selected to train the models and evaluate the performance on AD tasks. MNIST and FMNIST comprised of handwritten digits and pieces of clothing, respectively. Both of the datasets are widely

adopted as a benchmark for several AE models for UAD tasks. MNIST and FMNIST datasets both contain 60,000 training samples and 10,000 test samples having 28×28 gray-scale pixels with 10 classes.

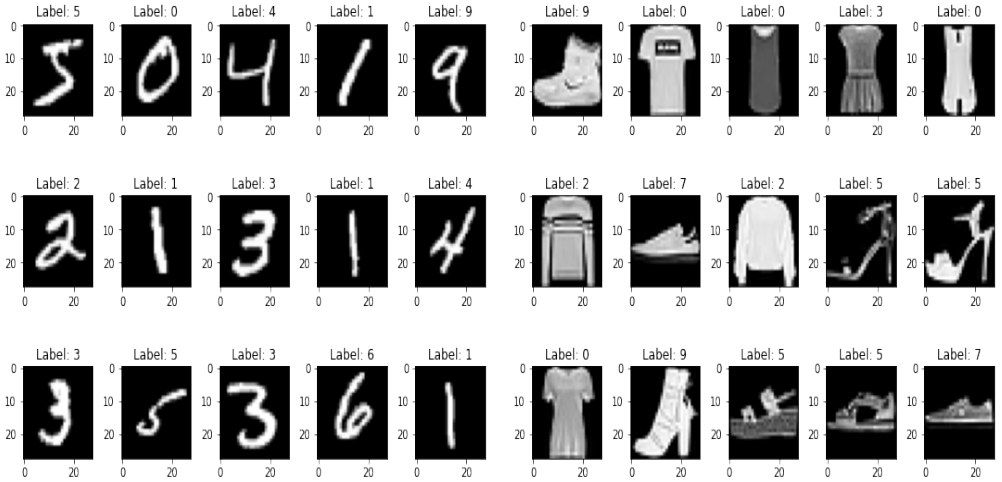


FIGURE 3.1: Example images from MNIST and FMNIST datasets

3.4.2 Evaluation Metric

To separate anomalous and standard samples, they are defined as negative and positive instances, respectively. Therefore, downstream the anomaly detection task is often referred to as a two-class classification problem. It is essential to identify the properties of the positive and negative categories. In general, the primary criteria to define anomalies is to calculate reconstruction error or error metrics such as MSE, binary-cross entropy, and z-score in conjunction with a loss function. Anomalous samples illustrate higher reconstruction errors compared to normal data samples. Based on the higher reconstruction error, anomalous and normal data can be classified.

Consequently, defining a threshold for either reconstruction error or accuracy metrics in the loss function is a primary step for accurate classification. Methods related to VAEs often overlook the importance of defining this threshold and emphasize the distribution of the data [96]. Considering the scope of this research, we took both reconstruction error and error metrics as a measurement of the anomalies, taking reference from the loss functions of AEs described earlier in the classifications of AEs (section 3.2). However, to create a similar baseline to compare the accuracy of the autoencoders, we utilized both the Average Precision Curve (AP)

and Receiver operator characteristic (ROC-AUC) as a single threshold evaluation metric alongside the loss function.

3.4.3 Platform Configurations

One of the major contributions of the study is to inspect the reproducibility of the existing autoencoder methods. In recent years, deep learning and neural network models have been heavily criticized for lacking reproducibility in different platforms [97]. Also, reproducing results incurs a high computational cost, which creates a significant barrier for other authors to replicate the codes for the algorithm. We gathered publicly available references from GitHub to replicate and reproduce the models. All codes taken from the GitHub repository was converted back to python 3.7.11 version file, which can either run on Windows 10 or any Linux system. Moreover, an environment configuration file with required python packages is provided by the authors[1] to easily re-configure in different platforms (e.g., python shell, Jupyter Notebook, or Google Colab machine). In Table 3.2, the configuration is provided for the machine where the coding and reproducibility tasks took place.

TABLE 3.2: Machine configuration

Machine Configuration	
CPU	Intel® Core (TM) i5-6200U CPU @ 2.30GHz - 2.40 GHz
GPU	Intel® HD Graphics 520 (2GB) and NVIDIA GeForce 920MX (2GB)
RAM	8.00 GB
OS	Windows 10, python 3.7.11

3.4.4 Challenges For Reproducibility

Reproducing any existing model is a challenging task. While we tried to implement the selected architecture as closely as intended by the original authors, we faced the following challenges:

- For officially available open-source repositories, we only converted the python version to 3.7.11. No changes were made to the original implementation.

[1] See this GitHub repository: `https://github.com/UMDimReduction/autoencoder-Efficiency_vs_Tradeoffs`.

However, there were often issues with deprecated modules or libraries. For such cases, we converted the function into the newer version. For example, loss functions of tensorflow-estimator are only available to use as API in version 2.1.0 directly. Therefore, every architecture using a newer version than 2.1.0 had to revert to 2.1.0.

- For the architecture with no publicly available reference implementation or repository, we reproduced the results taking the reference paper into account, and implemented the methods. Moreover, we also took partial references from resources mentioned in Table 3.4.

TABLE 3.3: Open source references for the selected architectures

Model	Link	API	Reference Type
DAE	`https://git.io/J15fg`	TensorFlow	Partial Reference
	`https://git.io/J15fP`	TensorFlow	Partial Reference
SAE	`https://git.io/J15v7`	PyTorch	Partial Reference
CAE	`https://git.io/J15vH`	TensorFlow	Partial Reference
VAE	`https://git.io/J15fY`	TensorFlow	Partial Reference
	`https://git.io/J15fW`	PyTorch	Official Repo
β-VAE	`https://git.io/J15J3`	PyTorch	Official Repo
adVAE	`https://git.io/J15fR`	PyTorch	Official Repo
CVAE	`https://git.io/J15vA`	PyTorch	Official Repo
IWAE	`https://git.io/J15fs`	Theano	Partial Reference
PAE	`https://git.io/J15fD`	TensorFlow	Official Repo
RDA	`https://git.io/JfYG5`	TensorFlow	Official Repo
VQ-VAE	`https://git.io/J15fT`	PyTorch	Official Repo

3.5 Results

In this section, we present the results of the experiments on the MNIST and FMNIST datasets described in the previous section. The results are divided into two parts: a qualitative part where we visually assess the reconstructed images, and a quantitative part where we assess the performance of each architecture on the anomaly detection task.

3.5.1 Training Time

Training time is a crucial part to determine the computational cost and ability to reproduce. Instead of using a resourceful GPU machine, the experiments were conducted on a general setup that most machine learning researchers can avail. Therefore, it is expected that the training time might fluctuate between 20%-25%, varying the hyperparameters in different reproductive configurations. The training times mentioned in Table 3.4 provide a generalized estimate of time to prepare the models using both MNIST and FMNIST.

TABLE 3.4: Training times for each architecture on both datasets

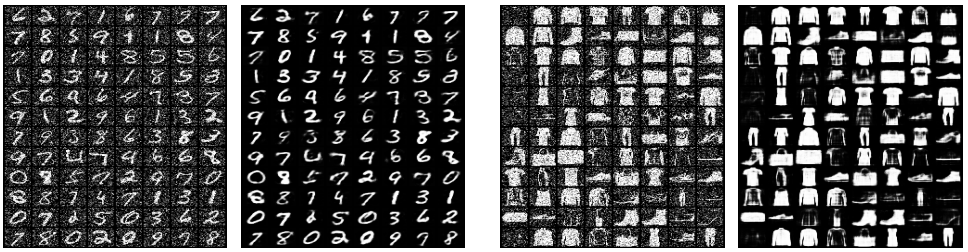
Model	Training Time (HH: MM: SS)	
	MNIST	FMNIST
DAE	0:47:00	0:56:45
SAE	1:07:21	1:45:10
CAE	0:13:25	0:22:27
VAE	0:35:50	0:42:45
β-VAE	0:23:33	0:55:27
adVAE	0:59:45	1:01:45
CVAE	0:35:21	0:59:40
IWAE	0:34:18	0:37:16
PAE	0:20:17	0:33:21
RDA	0:59:21	1:02:10
VQ-VAE	0:30:33	0:40:21

3.5.2 Reconstructing Images and Generating Samples

The reconstruction quality of the autoencoders is a key criterion to distinguish different methods. If the reconstruction quality is excellent, that algorithm primarily performs well in classification and anomaly detection.

3.5.2.1 DAE

The task of DAE is to reconstruct an image from a corrupted noisy image. The noise was added sequentially from 20% to 52%.

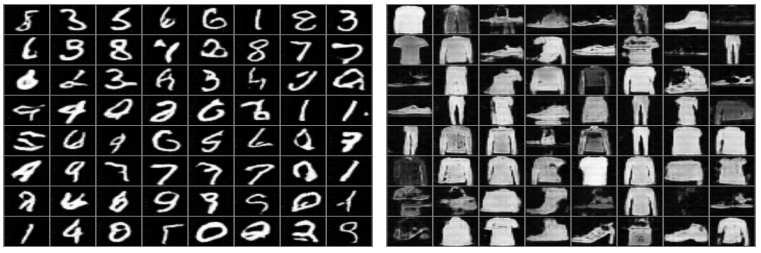


(A) (B) (C) (D)

FIGURE 3.2: Figure 3.2a and 3.2c illustrates the 27% noisy input data by MNIST and FMNIST respectively. Figure 3.2b and 3.2d is the reconstruction from the noisy input.

3.5.2.2 RDA

Similar to the DAE, the RDA task defines in such a way to generalize the DAE and where no clean, noise-free data is available. Using the l_1 or $l_{(2,1)}$ regularization, penalize the noisy data and reconstruct the original input.

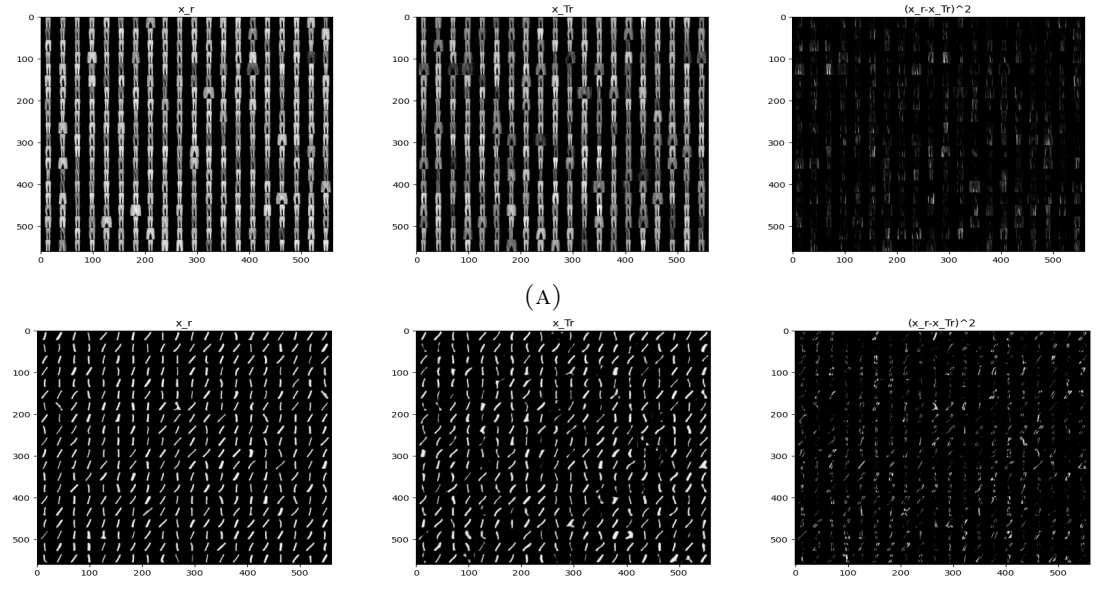


(A) (B)

FIGURE 3.3: Figure 3.3a and 3.3b is the 29% improved anomaly data reconstruction by using regularization observed from MNIST and FMNIST datasets, respectively. The leftmost column represents the original anomalous sample, followed by a reconstructed improved sample in the column afterward.

3.5.2.3 adVAE

The adVAE inspects the restoration of the original input using the Gaussian prior distribution. Adversarial learning predicts whether a sample comes from the hidden core of the autoencoder or the prior distribution.



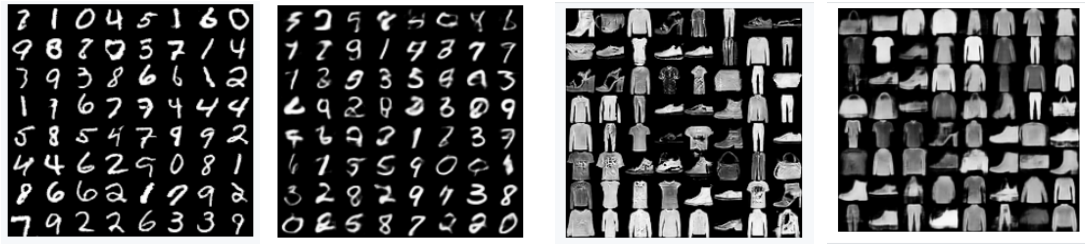
(A)

(B)

FIGURE 3.4: Figure 3.4a and 3.4a shows the reconstruction process on MNIST and FMNIST dataset respectively. Three steps accommodate the process: the first image shows the sample. The second image is a sample reconstruction. The final image is acquired after Gaussian transformer T is applied on reconstruction.

3.5.2.4 VAE

The experiment of VAE focuses on learning the latent representation. The latent representation regulates the easy reconstruction of the learned input data. VAE reconstructs samples using low construction probability from a corrupted input or anomalous sample.

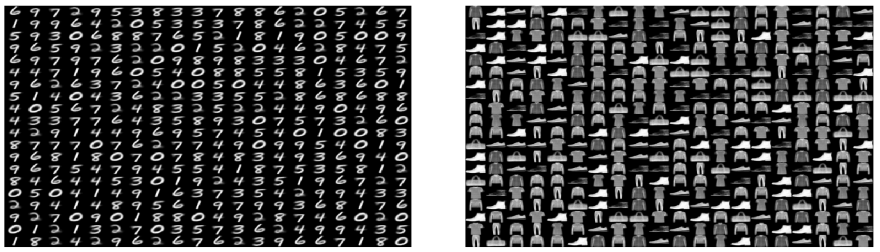


(A) (B) (C) (D)

FIGURE 3.5: Figure 3.5a and 3.5c represents some sample anomaly data from MNIST and FMNIST dataset. Similarly, Figure 3.5b and 3.5d shows the reconstruction from the sample anomaly data.

3.5.2.5 β-VAE

β-VAE emphasizes learning the disentangled representation. Based on the tuned value of β, the VAE and β-VAE are distinguished. While training the model, when the β value is 1, the objective function is simply VAE, but when the value is set to β >1, we obtain the β-VAE.

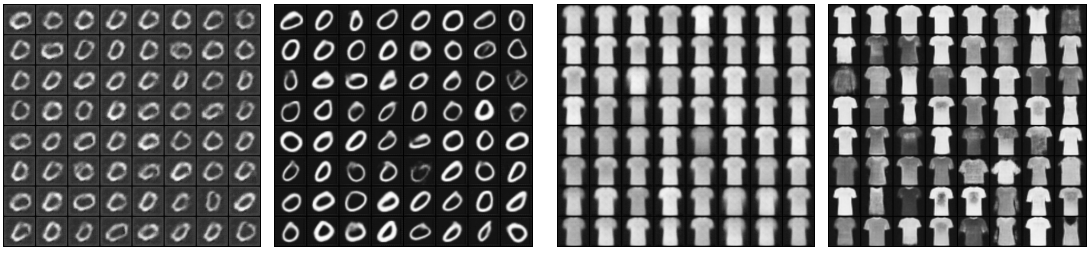


(A) (B)

FIGURE 3.6: Figure 3.6a and 3.6b shows randomly generated samples from MNIST and FMNIST dataset where β value varied from 1.5 to 2.

3.5.2.6 CVAE

CVAE is an extended method of VAE where the data generation is controlled by optimizing the variational lower bound. The experiment of CVAE consists of generating specific data points based on the label and reconstructing them based on $Q(z \mid x, c)$ distribution.

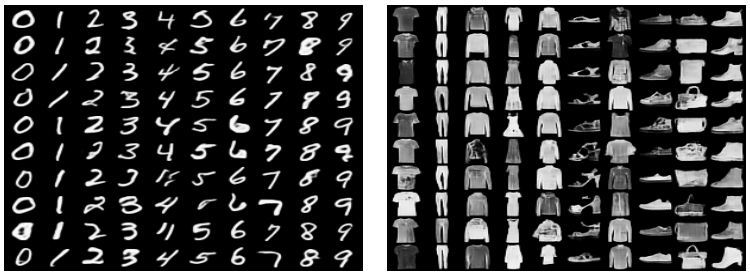


(A) (B) (C) (D)

FIGURE 3.7: Figure 3.7a and 3.7c represents the initial step (epoch-1) of sample generation from MNIST and FMNIST dataset respectively. On the other hand, Figure 3.7c and 3.7d shows the final output (epoch-50) of sample generation.

3.5.2.7 VQ-VAE

VQ-VAE is derived from the original VAE, where the learning of latent representation is discrete. The challenge of the experimental setup was to train discrete latent embeddings and visualize their importance. The reconstruction uses discrete network blocks, and the latent representation is acquired afterward. As a result, the latent representation and reconstructed image are more compacted and serrated than the original VAE.

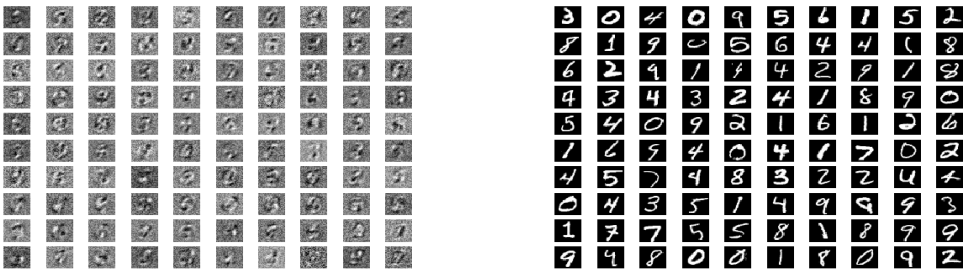


(A) (B)

FIGURE 3.8: Each column in both Figure 3.8a and 3.8b represents the reconstruction of anomalous samples from different classes from the MNIST and FMNIST datasets, respectively.

3.5.2.8 SAE

The SAE uses the "sparsity penalty" while reconstructing the original input. The sparsity penalty uses the l_1 regularization and minimizes the loss function. The primary intention of the SAE experiment is to visualize the learned weight from the input vector.

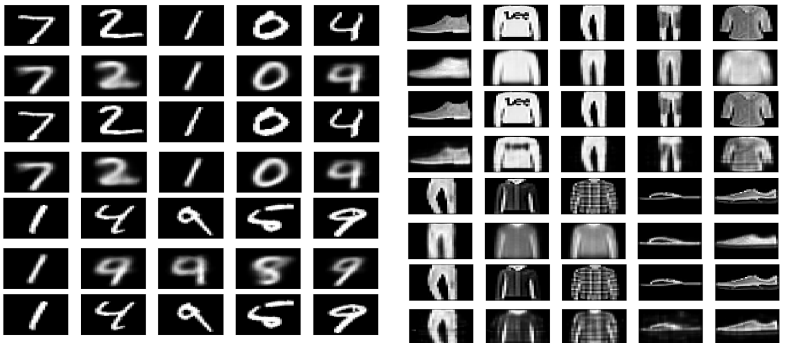


(A) (B)

FIGURE 3.9: Figure 3.9a represents the random sample with a sparsity of 0.45, and the reconstruction from that sparse input is illustrated in Figure 3.9b.

3.5.2.9 CAE

CAE is a deterministic autoencoder that adds a regularizer to the loss function. The intention of the regularizer is to make the encoding less sensitive to minor variances. The end result illustrates the learned representation is less sensitive towards the training input.

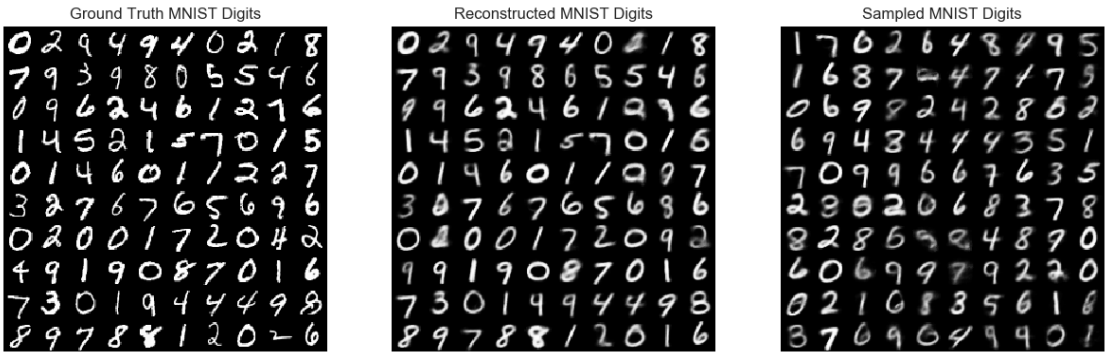


(A) (B)

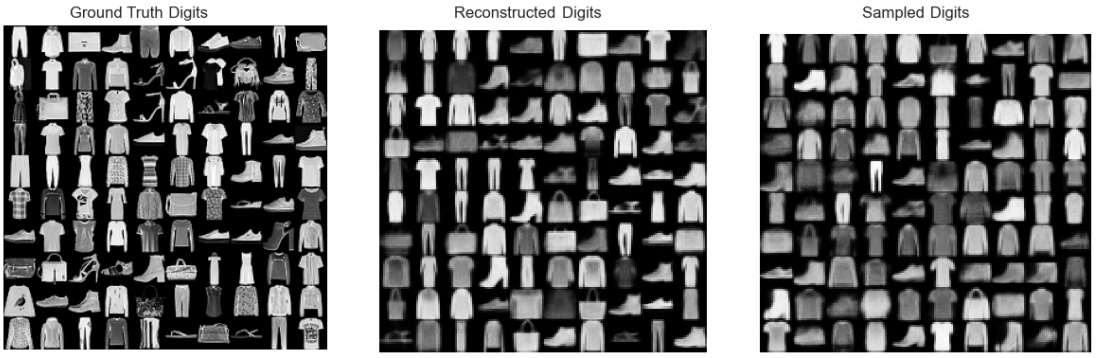
FIGURE 3.10: Both Figure 3.10a and 3.10b shows the anomalous data samples reconstruction on MNIST and FMNIST dataset using 2-depth deterministic CAE.

3.5.2.10 IWAE

The IWAE focuses on deriving the better lower bound on the marginal likelihood by minimizing the cost. The cost function weights the re-sampling in the ELBO. The derivation provides the importance sampling in ELBO by the distinction acquired from a mini-batch of the sample from samples and reconstructions.



(A)

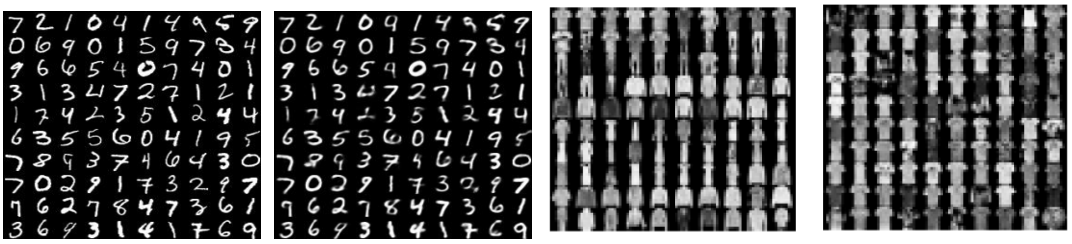


(B)

FIGURE 3.11: Using two stochastic layers and $k = 50$, Figure 3.11a and 3.11b represent the sample generation from MNIST and FMNIST dataset. In both Figures, the left, middle and right image presents the "ground truth sample," "reconstructed sample," and "mini-batch samples," respectively.

3.5.2.11 PAE

The PAE is the minimalist generative autoencoder that targets to achieve minor reconstruction errors. Besides the minimal error, the PAE is easy to train in low hyper-parameter tuning to reach lower reconstruction error and less training time. A similar reconstruction like the VAE or RDA is showed in Figure 3.12, but it occupied using significantly fewer hyperparameters and training time.



(A) (B) (C) (D)

FIGURE 3.12: Figure 3.12a and 3.12c represents the samples from MNIST and FMNIST dataset whether Figure 3.12b and 3.12d shows the reconstruction using sampling quantities of $k = 50$.

3.5.3 Latent Space Representation

Latent space representation enables lower-dimensional views from a high-dimensional presentation. Each point in the images presented in Figure 3.13 and 3.14 is the latent code from the test set, and different colors are used to distinguish digits. The x and y-axis scales vary due to the models' representation but do not affect representation.

3.5.4 Interpolation and Manifold

In order to visualize the high-density regions, the task of extracting 2-D manifold and interpolation is efficient for autoencoders. After the latent space representation task, the manifold and interpolation show the most populated regions. If this representation accurately describes the original data, we can conclude that the corrupted input is effectively canceled out. Based on the interpretation, the latent space representation can be evaluated as well (MNIST- Figure 3.15 and FMNIST Figure 3.16).

3.5.5 Quantitative Comparison

Along with comparing the results with qualitative observations such as 2-D manifold, latent space representation, and sample reconstruction, we also demonstrated ROC-AUC for each method. Table 3.5 reports the result of evaluation metrics

reconstructed. Since ROC-AUC is not reported on the references paper, the numerical results mentioned in this table might be inconsistent with the referenced result. However, the quantitative comparison stated in Table 3.5 provides the distinguishable ground to compare autoencoders for the downstream tasks of UAD.

TABLE 3.5: Results of ROC-AUC for all architectures on both datasets

Model	MNIST	FMNIST
DAE	*0.73*	0.56
SAE	0.83	0.66
CAE	0.81	*0.22*
VAE	0.78	0.61
β-VAE	0.87	0.59
adVAE	**0.93**	**0.87**
CVAE	0.80	0.66
IWAE	0.87	0.57
PAE	0.89	0.64
RDA	0.82	0.60
VQ-VAE	0.82	0.56

3.6 Discussion

In these experiments, we focused on three evaluation criteria: 1) Reconstruction and generation of sample inputs; 2) Latent space representation; and 3) Manifold learning and interpolation. Along with these three comparisons, one of the core focus of this study is the efficiency vs. trade-offs hypothesis that is mentioned in section 3.6.1. The ability to reconstruct and generate samples are the two most common ground to differentiate autoencoders for the task of UAD.

If we consider noisy input, DAE performs better reconstruction if the noise is less than 25%. As we increase the noisy inputs, SAE performs better than DAE. However, DAE and SAE seem to have a complication in 5 and 9 reconstructions. On the other hand, RDA barely can reconstruct the noisy inputs; the digits 6, 4, 2, and 9 are similarly reconstructed.

Among variational methods (VAE, β-VAE, VQ-VAE, CVAE), VQ-VAE and CVAE produce sharper reconstructions. For the former, this is likely due to the discrete latent space. Compared to VQ-VAE, β-VAE generates more accurate and sharp

pixel outputs. On the other hand, VAE introduces noise while sampling but minimizes it efficiently. β-VAE tends to find additional latent factors and learn the characterization of disentanglement than the other models.

For the other methods (PAE, CAE, IWAE), we observe the abnormality in reconstructing some specific digits like 6, 9, 2, and 4. CAE shows blurry reconstruction, but they are interpretable, IWAE has less blurry reconstructions, but the interpretation is unclear. PAE has both clear and sharp reconstructions, but some reconstructions (e.g. 5, 8, and 9) are somewhat inaccurate.

Interestingly, there are few similarities in latent space representation both in MNIST and FMNIST datasets. If we ignore the rotation of clusters, VAE, SAE, CAE, and PAE separate the digits 0, 5, 6, and 7 more efficiently for MNIST. For FMNIST, IWAE, VAE, and β-VAE, CVAE show some similarities while presenting clustered labels, but most of them are not clearly distinguished. We can derive the following observations from the latent space visualizations:

- For both datasets, the latent representation of RDA, CVAE overlaps with most of the sample. One of the reasons for RDA can be the split input with two layers (Deep Autoencoder and Noisy input).
- β-VAE, VAE, VQ-VAE have some apparent similarities between the plots in MNIST. The clusters of zero, three, and seven are nicely separated. In both cases, the latent layer seems to encode the structural similarities. Also, the plots of VAE and β-VAE show the influence of the KLD loss as the latent occupy less space than the other methods by restricting the samples of a standard Gaussian.
- SAE, PAE, CAE has the most efficient way to separate the clusters (MNIST). SAE and PAE both contain the Weight decay parameter that equally distributes the denoising data. IWAE and adVAE also illustrate similar properties but are not efficient in separate clusters like the others.

2-D manifolds and interpolation show latent space interpretation more accurately than reconstruction. Most anomalous data are re-generated along with the manifold to evaluate reconstruction and generation techniques for different architectures. For both MNIST and FMNIST, similar to latent space, the VQ-VAE illustrates the

KLD loss effect in manifold structure. Surprisingly, β-VAE shows no outcome of disentangled latent factors in latent space. However, for the 2-D manifold, β-VAE shows the lenient latent factors. One possible interpretation for such behavior is the rotation of each digit in terms of coloring; if we reroute the digits to their original axis (similar to β-VAE in the FMNIST dataset), the effect of showing more substantial latent factors can be observed.

- For sample generation, we only considered adVAE, VAE, VQ-VAE, SAE, β-VAE and eliminated the other autoencoders. The elimination is to narrow down the generative models and only compare them with the typical ground architecture.
- The variance of samples is relatively small. Only the β-VAE generated the most satisfactory sample. The β-VAE has the highest emphasis on discharging Gaussian prior. Therefore, the encoder output is the most similar to standard Gaussian samples, and the decoder generates the same latent code sampled from a true standard Gaussian. On the other hand, the construction of CVAE and VQ-VAE is a bit surprising as they resemble the MNIST digits poorly.
- The rest of the autoencoders generates average samples with a fade-out and noisy pixels. In some cases (SAE and adVAE), they are bend out due to similar class distributions.

3.6.1 Efficiency and Trade-offs

All the results and discussions mentioned above lead towards formulating our hypothesis: efficiency vs. trade-offs in experimented autoencoders. Table 3.6 sets the background to lead towards our conclusion as follows:

3.7 Conclusion

In summary, we conclude that:

- General variational methods are more efficient when enforcing special conditions to control discrete latent representation. For example, the performance of VQ-VAE, β-VAE, and CVAE is superior to that of the original VAE, as VQ-VAE uses the auto-regressive prior to pair representations, while β-VAE emphasizes locating disentangled latent factors.

- A better reconstruction from noisy data can be obtained by removing unwanted noisy corrupted input if we introduce a hidden activation layer to activate a specific layer at a time and improve robustness, avoid over-fitting of the data. For example, SAE has a sparse hidden layer unlike DAE or RDA, and the generated output from SAE has better pixels and classification accuracy than the other ones.

- Adversarial and generative methods model arbitrary latent space that imposes a prior that may not fit the data distribution well. Therefore, the adversarial generator networks with precious structural latent space sampling adaptability and the ability to normalize the space are more efficient. In particular, learning the positive and negative latent representations to avoid over-smoothing is the key for such types of autoencoders.

Although this study provides a brief overview and experiments of AE architecture, including recent extensions of such methods including InfoVAE [5], Clustering VAE [98],VFAE [6], DIP-VAE [7], FactorVAE [3], β-TCVAE [3], RFVAE [59], extensions of GANs [67–69], and extensions of feature extraction methods [37, 61] will enhance this literature with the further foundation for VAE architectures.

3.7.1 Scope and Future Research Directions

The conclusion from the efficiency and trade-off study perceives few vital research directions for VAEs. Firstly, we drew an overview of qualitative and quantitative results from different AEs based on reconstruction quality, sample generation, latent space visualization, and anomaly detection accuracy. Then, based on the empirical evidence, we derive the efficiency and trade-off observations. We find the following two crucial avenues of improvement for VAE architectures: efficiently enforce disentangled factors for retaining valuable information and learning structured distribution to combine posterior variants of VAEs.

Based on these two avenues, we can discuss future research on VAEs into the following three general categories:

- **Posterior distribution**: Understanding the principle of posterior loss and introducing the complete disentangled feature within VAEs by regularizing the posterior distribution. Finding the proper way to resolve this issue will improve the quality of generated samples from autoencoders, especially VAEs.
- **Structured distribution**: The structured prior distribution is crucial for accomplishing better representation clustering of data. VAEs in some data samples generate blurry and erroneous output; this can be tackled by imposing Gaussian Mixture Model (GMM) prior and the architecture of Vectorized Quantization (VQ). Moreover, GMM prior can also improve the variational optimization while decoding the information bottleneck.
- **Models Architecture**: Combining and enhancing different AE architecture to improve bottleneck decoding is the most acceptable way to tweak the AEs. e.g., VAEs or CVAEs combined with PAE can decrease the blurred effect on the generated samples.

Address the categories mentioned above; future research on VAEs can be effectively utilized in applications like detecting serious diseases using Bio-signal applications, speech processing, image generation, and controlling data distributions.

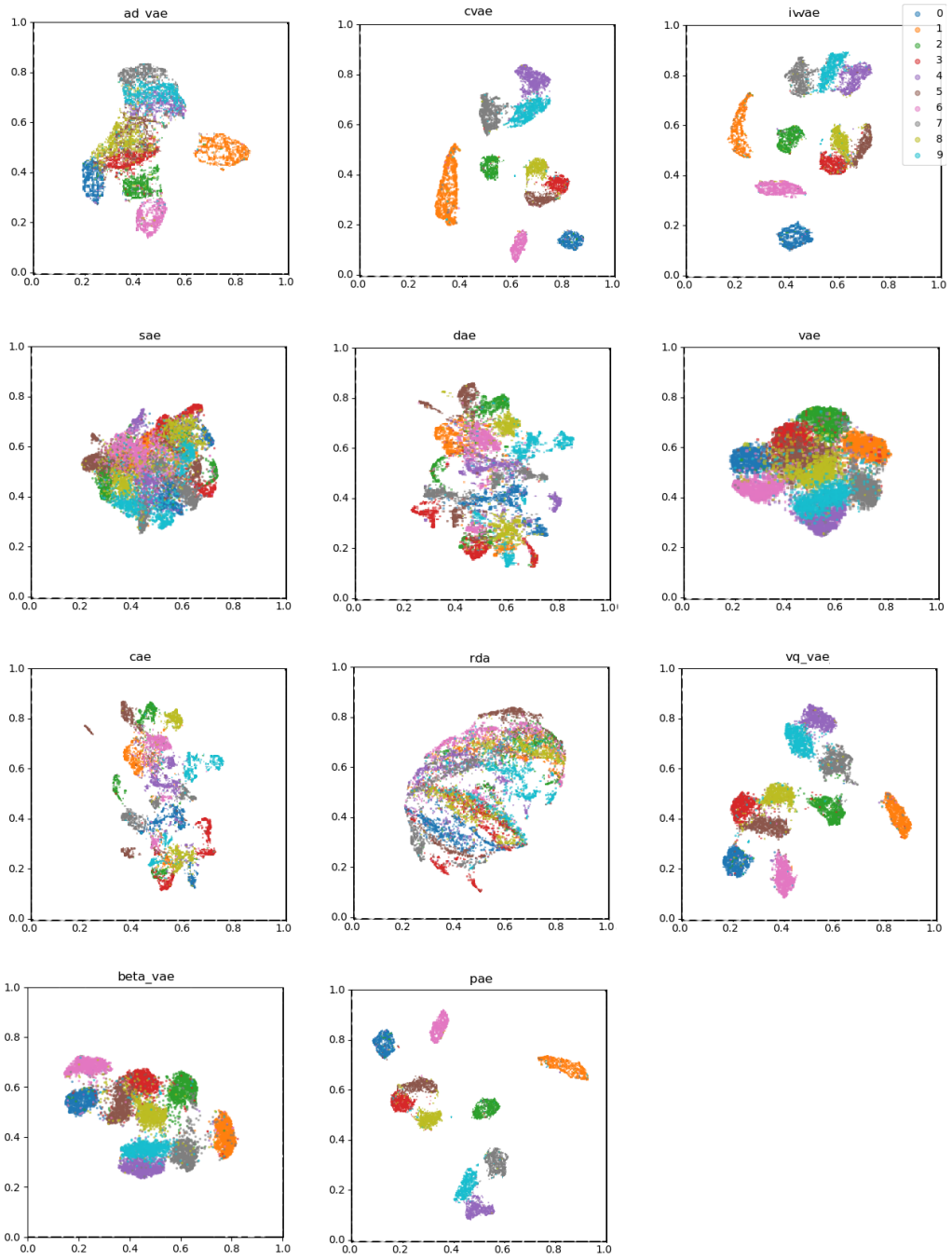


FIGURE 3.13: Latent Space Representation (MNIST)

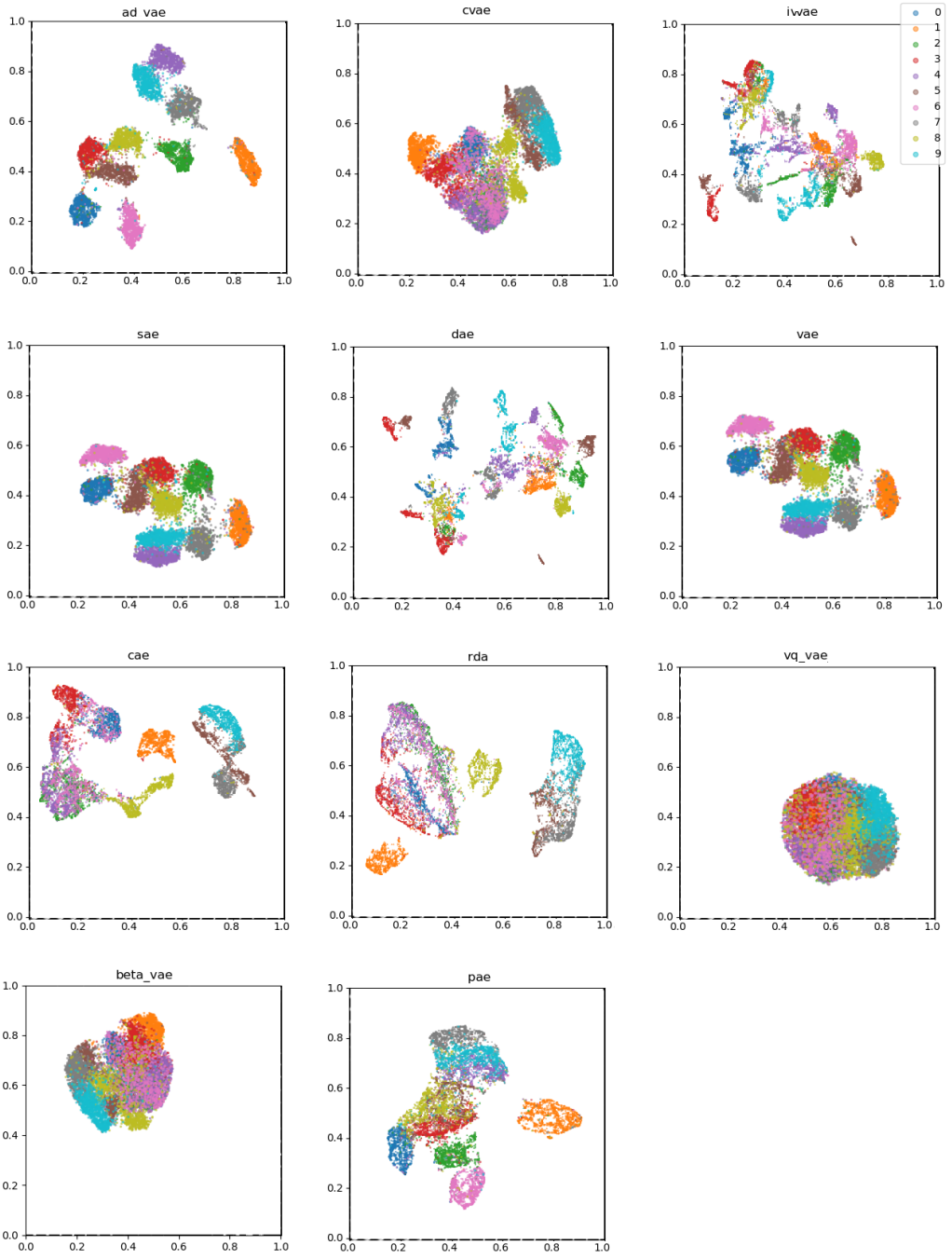


FIGURE 3.14: Latent Space Representation (FMNIST)

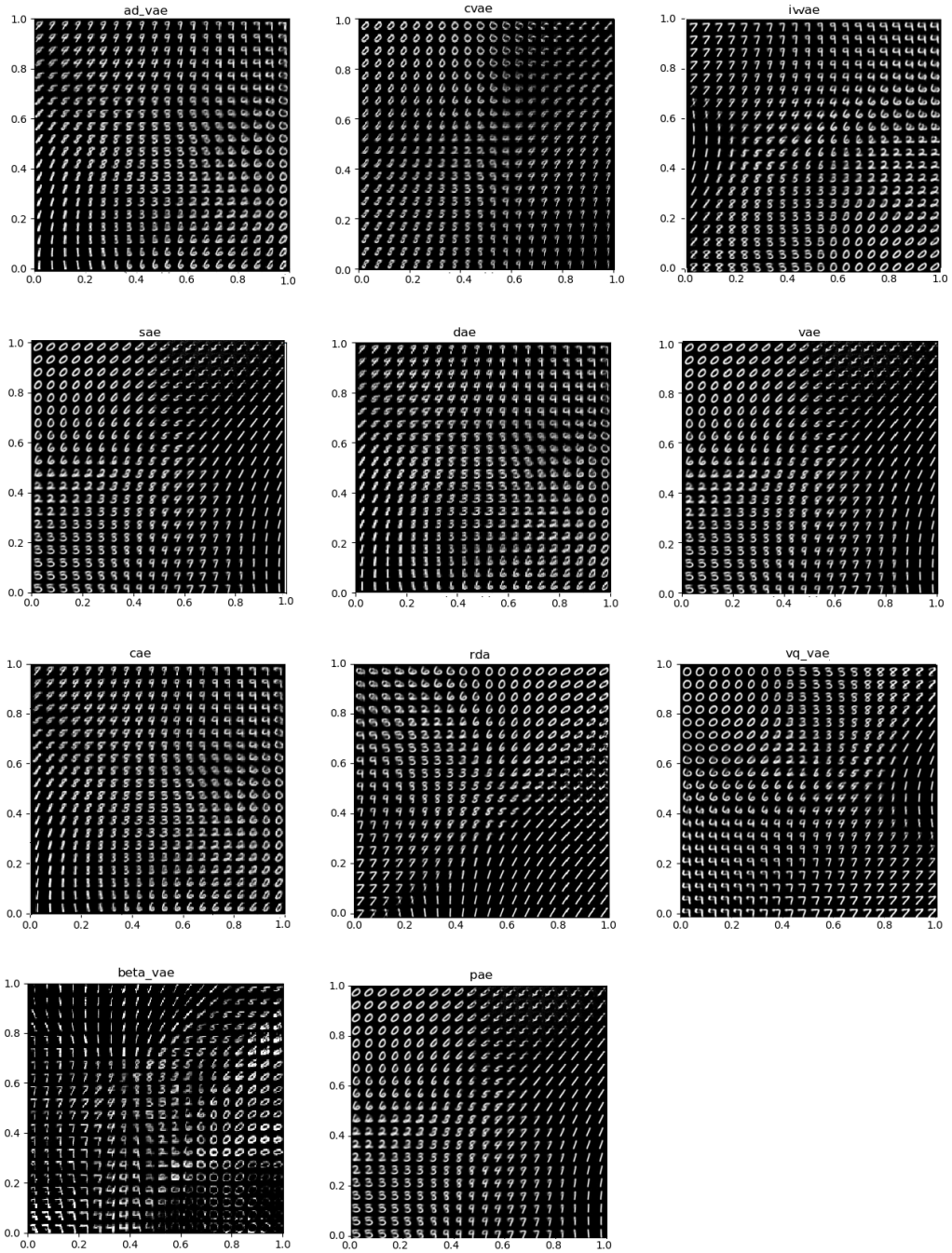


FIGURE 3.15: 2-D Manifold (MNIST)

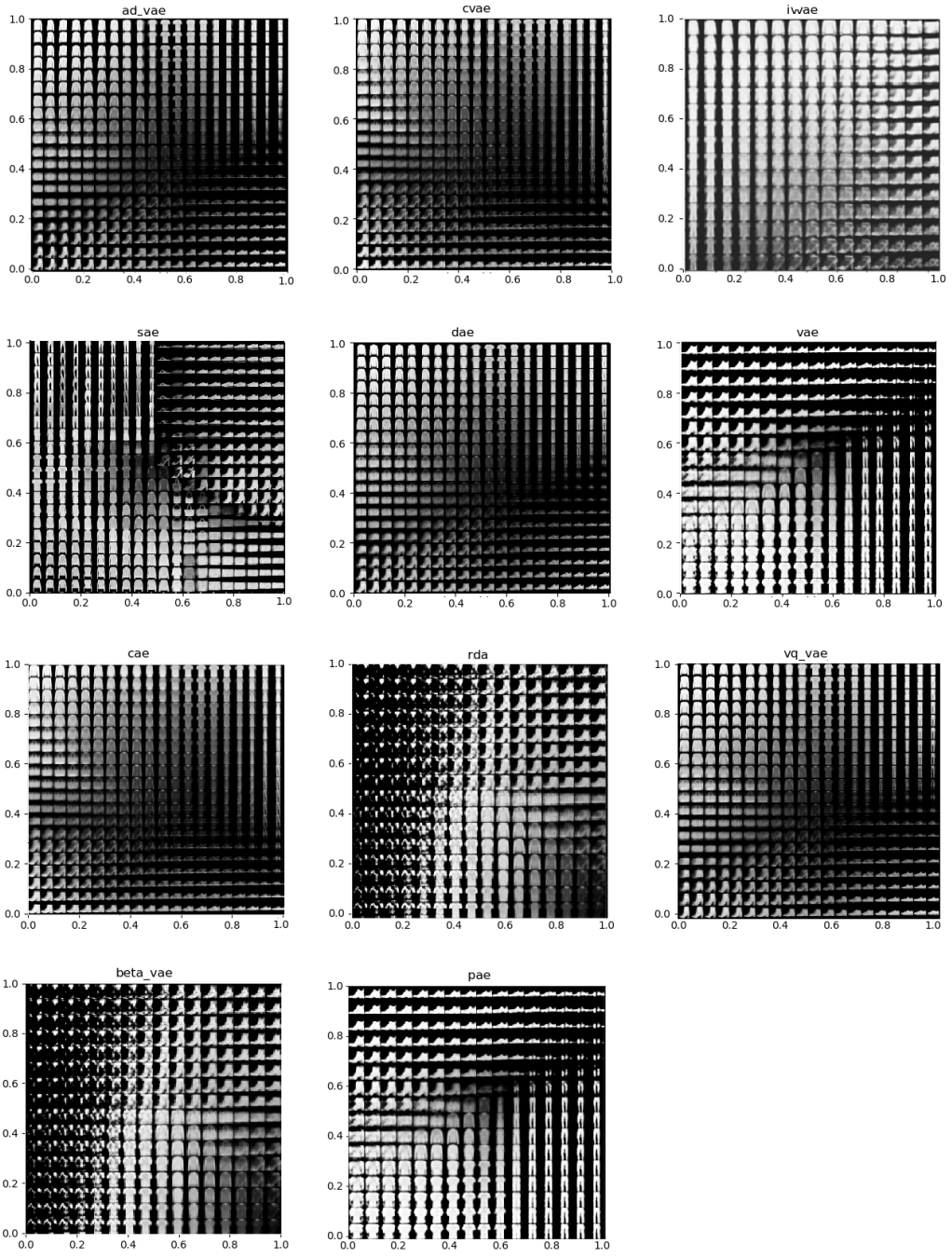


Figure 3.16: 2-D Manifold (FMNIST)

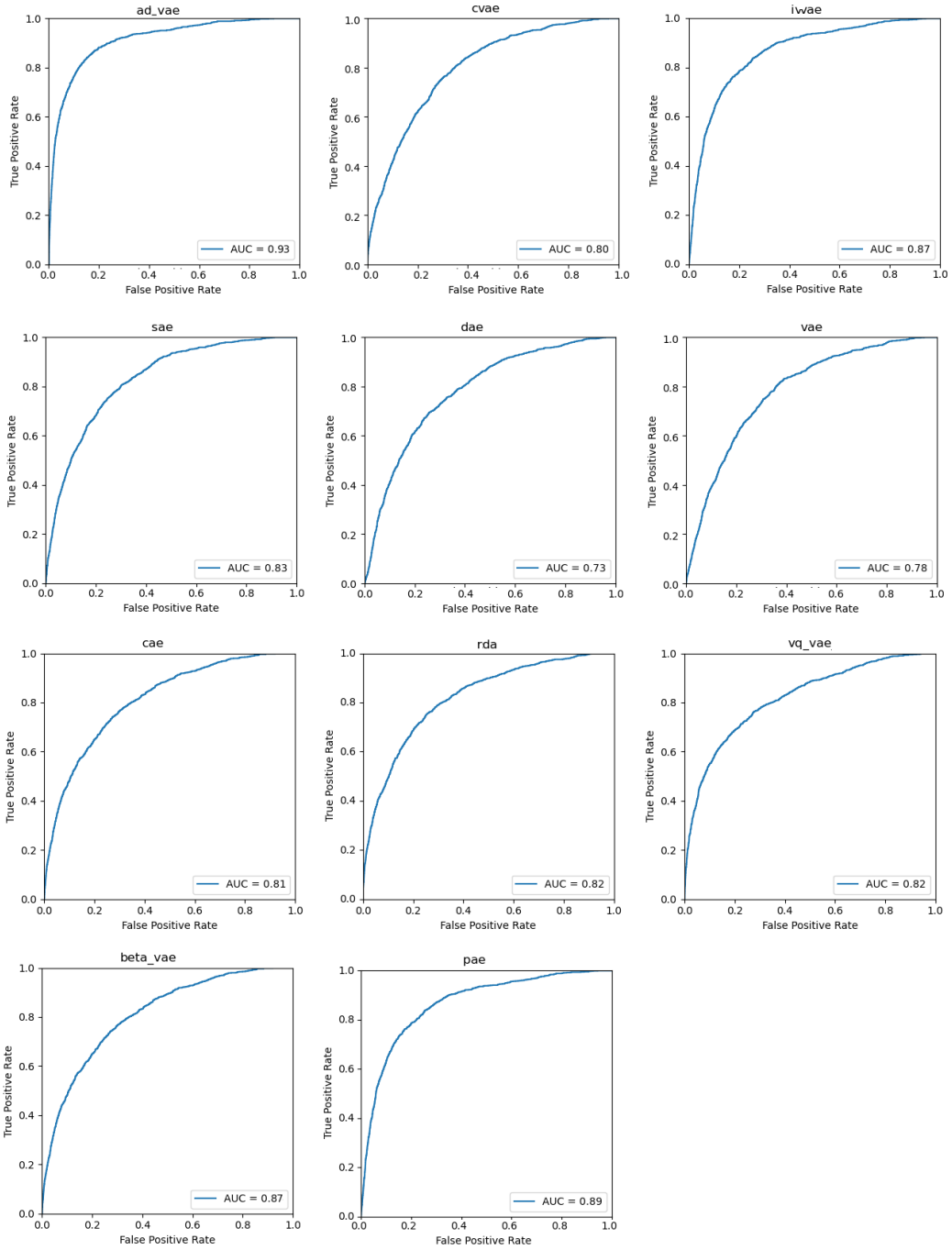


FIGURE 3.17: ROC-AUC (MNIST)

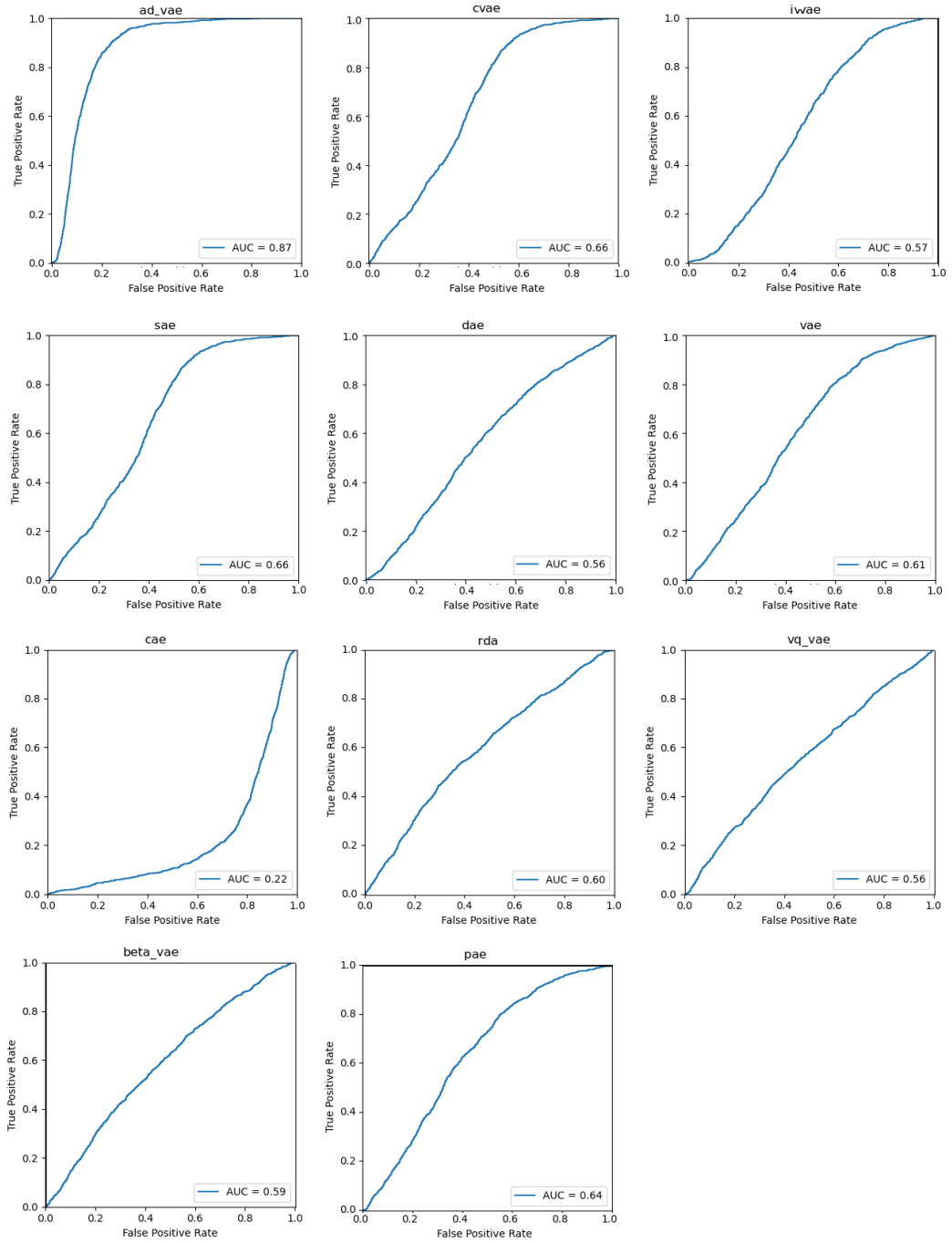


FIGURE 3.18: ROC-AUC (FMNIST)

TABLE 3.6: Efficiency and Trade-off Comparison

Autoencoder	Efficiency	Trade-offs
DAE	• Efficient in correcting corrupted input.	• More sensitive to noise when more than 45% input is corrupted.
	• Extracting and selecting robust features after denoising the input.	• Too much lossy construction and hard to interpret the latent space.
RDA	• With deep layers, the RDA can both denoise the input and classify data effectively.	• No clear interpretation in latent space.
	• Splitting the reconstructed input data noise or outlier data improves the robustness of the standard deep autoencoders.	• Poor and blurry reconstructed and generated sample compared to VAE and adversarial models.
adVAE	• Compared to VAE, the adVAE doesn't introduce any deterministic bias for optimizing the lower bound on the log-likelihood, as a result, it generates less blurry samples.	• For some training samples, the encoder/decoder fails to adapt the latent space to match the prior distribution.
	• adVAE is more efficient in introducing discrete latent variables.	• If the discriminator variable is not accurately perceived, the loss can spike unpredictably.
CVAE	• Combine VAE and output prediction using a conditioned variable.	• Due to the conditioned variable, the computational complexity is too high.
	• Effective in improving classification Accuracy using the labeled condition.	• To distinguish latent space, the testing sample requires more data.
	• Comparatively scalable in respect to other conditional variational models.	
VAE	• Incorporates the probabilistic distribution with generative modeling.	• The curse of dimensionality limitation in Gaussian prior distribution.
	• Improves the unsupervised models in generating samples.	• Constructed sample tends to get blurry.
	• Learning smooth latent representation and visualizing latent space.	• Latent representation does not show interpretable results.

β-VAE	• Combing the VAE features for more efficient learning.	• For tasks like classification and disentangled representation, there is no clear correlation.
	• Capable of automatically learning the disentangled representation.	• Might ignore latent variable or under-fitting model.
	• More scalable than the VAE and easy to train.	
VQ-VAE	• Removes the noise and corrupted details.	• Exhibit high entropy and very low interpretation from a latent variable.
	• Combing the VAE features for discrete representation learning.	• Not stable in complex sampling setup.
CAE	• Improve reconstruction of the generative models using a penalty term	• Penalty in a contractive layer can result in useless construction.
	• Efficient in capturing the local directions of variations on the data.	• CAE calculates the local information, hence may face lower latent reconstruction.
IWAE	• Using the same architecture of VAE, IWAE more accurately identifies tighter log-likelihood lower bound.	• Require fine-tuning based on the training data.
	• Compared to VAE, more effective in learning latent space representations, improving test log-likelihood.	• The accuracy of the classification heavily depends on the approximate posterior.
SAE	• Compared to DAE, the SAE is more efficient in denoising input.	• L1 regularization can fail where it is robust to outliers.
	• Sparsity condition in activation layers enables the control of the dependency of features.	• A small number of layers may fail to decode the complex latent space representation.
	• As it activates layers based on the condition, for certain uses it is very scalable.	
PAE	• Without parameter tuning, the PAE achieves a small reconstruction error than the VAE.	• Can wrongly estimate the reconstruction error.
	• Easy to train and more scalable.	• Require more metrics to verify the classification.

Chapter 4

Disentangled Conditional Variational Autoencoder for Unsupervised Anomaly Detection

Preamble to Manuscript 2.

While working on the first manuscript, I experimented with several architectures of VAE with different baseline comparisons, i.e., reconstruction, sample generation, classification of anomalies, ROC-AUC, and latent space visualization. Visualization of the latent variables and quantitative results from anomaly classification highlighted a few important observations. Firstly, we observed blurry reconstruction from the original VAE frameworks, which were resolved using the adVAE. Among different criteria of evaluation, we observed that the following two properties hugely improve anomaly detection:

- Minimizing information loss in posterior-variants of VAE (i.e., β-VAE, VQ-VAE) leads to improving the reconstruction quality.
- The nearer we minimize the negative ELBO score, the better it gets in anomaly detection accuracy.

However, such an interpretation is intractable without experimenting with several loss functions by either modifying the loss function or proposing a new loss and objective function. Around the same time, I started implementing different VAE architectures and tested their accuracy in the MNIST dataset. In total, I implemented 31 architectures of autoencoder using both PyTorch and TensorFlow packages. As I started training these

large models, I faced reproducibility issues and little or no resources with verifying the implementation. I greatly suffered from compatibility issues with Python and TensorFlow, such as: deprecated packages or libraries and version compatibility with python 3.6.3 and TensorFlow 2.1. After facing such issues, I took the issue of reproducibility as a part of my experiments and created a python implementation to reproduce the results easily. Going forward, while reading the meta-priors literature and multivariate information theory, I started introducing the regularization of loss function using the CorEx techniques. Meanwhile, I worked on finding appropriate datasets for testing the different methods. After several months of training the models and testing them in a different context, I chose several MNIST-type datasets to illustrate the results. Thus, this manuscript focuses on both dCVAE experiments and the issue of reproducibility. Currently, this manuscript is under review at the *Eleventh International Conference on Learning Representations* (ICLR 2023).

Disentangled Conditional Variational Autoencoder for Unsupervised Anomaly Detection

Asif Ahmed Neloy
Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada

Maxime Turgeon
Department of Statistics
University of Manitoba
Winnipeg, MB, Canada

Abstract

Recently, generative models have shown promising performance in anomaly detection tasks. Specifically, autoencoders learn representations of high-dimensional data, and their reconstruction ability can be used to assess whether a new instance is likely to be anomalous. However, the primary challenge of unsupervised anomaly detection (UAD) is in learning appropriate disentangled features and avoiding information loss, while incorporating known sources of variation to improve the reconstruction. In this paper, we propose a novel architecture of generative autoencoder by combining the frameworks of β-VAE, conditional variational autoencoder (CVAE), and the principle of total correlation (TC). We show that our architecture improves the disentanglement of latent features, optimizes TC loss more efficiently, and improves the ability to detect anomalies in an unsupervised manner with respect to high-dimensional instances, such as in imaging datasets. Through both qualitative and quantitative experiments on several benchmark datasets, we demonstrate that our proposed method excels in terms of both anomaly detection and capturing disentangled features. Our analysis underlines the importance of learning disentangled features for UAD tasks.

4.1 Introduction

Unsupervised anomaly detection (UAD) has been a fertile ground for methodological research for several decades. Recently, generative models, such as Variational Autoencoders (VAEs) [54] and Generative Adversarial Networks (GANs) [69, 99], have shown exceptional performance at UAD tasks. By learning the distribution of normal data, generative models can naturally score new data as anomalous based on how well they can be reconstructed. For a recent review of deep learning for anomaly detection, see [100].

In a complex task like UAD, disentanglement as a meta-prior encourages latent factors to be captured by different independent variables in the low-dimensional representation. This phenomenon has been on disply in recent work that has used representation learning as a backbone for developing new VAE architectures. Some of the methods proposed new objective functions [2, 73], efficient decomposition of the evidence lower bound (ELBO) [3], partitioning of the latent space by adding a regularization term to the mutual information function [5], introducing disentanglement metrics [4], and penalizing total correlation (TC) loss [74]. Penalized TC efficiently learns disentangled features and minimizes the dependence across the dimension of the latent space. However, it often leads to a loss of information, which leads to lower reconstruction quality. For example, methods such as β-VAE, Disentangling by Factorising (FactorVAE) [4], and Relevance FactorVAE (RFVAE) [59] encourage more factorized representations with the cost of either losing reconstruction quality or losing a considerable among of information about the data and drop in disentanglement performance. To draw clear boundaries between an anomalous sample and a normal sample, we must minimize information loss.

To address these limitations, we present Disentangled Conditional Variational Autoencoder (dCVAE). Our approach is based on multivariate mutual information theory. Our main contribution is a generative modeling architecture which learns disentangled representations of the data while minimizing the loss of information and thus maintaining good reconstruction capabilities. We achieve this by modeling known sources of variation, in a similar fashion as Conditional VAE [60].

Our paper is structured as follows. We first briefly discuss related methods (Section 4.2), draw connection between them, and present our proposed method dCVAE (Section 4.3). In Section 4.4, we discuss our experimental design including competing methods, datasets, and model configuration. Finally, experimental results are presented in Section 4.5, and Section 4.6 concludes this paper.

4.2 Related Work

In this section, we discuss related work on autoencoders. We focus on two types of architecture: extensions of VAE enforcing disentanglement, and architectures based on mutual information theory.

4.2.1 β−VAE

β-VAE and its extensions proposed by Higgins et al. [2], Chen et al. [3], Mathieu et al. [73] is an augmentation of the original VAE with learning constraints of β applied to the objective function of the VAE. The idea of including such a hyper-parameter is to balance the latent channel capacity and improve the reconstruction accuracy. As a result, β-VAE is capable of discovering the disentangled latent factors and generating more realistic samples while retaining the small distance between the actual and estimated distributions.

Recall the objective function of VAE proposed by Kingma and Welling [54]:

$$L_{\text{VAE}}(\theta, \phi) = -\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x} \mid \mathbf{z}) + D_{\text{KL}}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p_\theta(\mathbf{z})\right). \tag{4.1}$$

Here, $p_\theta(\mathbf{x} \mid \mathbf{z})$ is the probabilistic decoder, $q_\phi(\mathbf{z} \mid \mathbf{x})$ is the recognition model, KLD is denoted by $D_{\text{KL}}(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p_\theta(\mathbf{z} \mid \mathbf{x}))$ parameterized by the weights (θ) and bias (ϕ) of inference and generative models. As the incentive of β-VAE is to introduce the disentangling property, maximizing the probability of generating original data, and minimizing the distance between them, a constant δ is introduced in the objective VAE to formulate the approximate posterior distributions as below:

$$\max_{\phi,\theta} \mathbb{E}_{\boldsymbol{x}\sim \text{X}}\left[\mathbb{E}_{q_\phi(z|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x} \mid z)\right]\right] \quad \text{such that} \quad D_{\text{KL}}\left(q_\phi(z \mid \boldsymbol{x}) || p(z)\right) < \delta. \tag{4.2}$$

Rewriting the Equation in Lagrangian form and using the KKT conditions, Higgins et al. [2] derive the following objective function:

$$\mathcal{L}_{\beta VAE}(\theta, \phi) = \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x \mid z)\right] - \beta D_{\text{KL}}\left(q_\phi(z \mid x) \| p(z)\right), \tag{4.3}$$

Here, β is the regularization coefficient that enforces the constraints to limit the capacity of the latent information z. When $\beta = 1$, we recover the original VAE. Increasing the value of $\beta > 1$ enforces the constraints to capture disentanglement. However, Hoffman et al. [101] argue that with an implicit prior, optimizing the regularized ELBO is equivalent to performing variational expectation maximization (EM).

4.2.2 FactorVAE

Disentangling by Factorising or FactorVAE is another modification of β-VAE proposed by Kim and Mnih [4]. FactorVAE emphasizes the trade-off between disentanglement and reconstruction quality. The authors primarily focused on the objective function of the VAE and β-VAE. The authors propose a new loss function to mitigate the loss of information that arise while penalizing both the mutual information and the KLD to enforce disentangled latent factors.

According to Hoffman and Johnson [102] and Makhzani and Frey [103], the objective function of β-VAE can be further extended into:

$$\mathbb{E}_{p_{\text{data}}\,(x)}[KL(q(z \mid x)\|p(z))] = I(x;z) + KL(q(z)\|p(z)), \tag{4.4}$$

Here, $I(x;z)$ is the mutual information between x and z under the joint distribution $p_{\text{data}}\,(x)q(z \mid x)$. FactorVAE learns the second term of $KL(q(z)\|p(z))$ and resolved the aforementioned issues by introducing total correlation penalty and density-ratio trick to approximate the distribution $\bar{q}(z)$ generated by d samples from $q(z)$. The loss function of the FactorVAE is as follows:

$$\begin{aligned}\mathbb{E}_{q(z|x^{(i)})}\left[\log p\left(x^{(i)} \mid z\right)\right] - KL\left(q\left(z \mid x^{(i)}\right)\|p(z)\right) \\ - \gamma KL(q(z)\|q(z))\end{aligned} \tag{4.5}$$

4.2.3 The principle of total Correlation Explanation (CorEx)

Gao et al. [74] introduced CorEx to mitigate the problem of learning disentangled and interpretable representations in a purely information-theoretic way. In general, for VAE, we assume a generative model where $\mathbf{x}$ is a function of a latent variable $\mathbf{z}$, and afterward maximize the log likelihood of $\mathbf{x}$. On the other hand, CorEx follows the reverse process where $\mathbf{z}$ is a stochastic function of $\mathbf{x}$ parameterized by θ, i.e., $p_\theta(\mathbf{z} \mid \mathbf{x})$, and seek

to estimate the joint distribution $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{z} \mid \mathbf{x})p(\mathbf{x})$. The underlying true data distribution maximizes the following objective:

$$\begin{aligned}\mathcal{L}(\theta; \mathbf{x}) = & \underbrace{TC_\theta(\mathbf{x}; \mathbf{z})}_{\text{informativeness}} - \underbrace{TC_\theta(\mathbf{z})}_{\text{(dis)entanglement}} \\ & = TC(\mathbf{x}) - TC_\theta(\mathbf{x} \mid \mathbf{z}) - TC_\theta(\mathbf{z}).\end{aligned} \tag{4.6}$$

Recall the definition of the total correlation (TC) in terms of entropy $H(\mathbf{x})$ [104]:

$$TC(\mathbf{x}) = \sum_{i=1}^{d} H(\mathbf{x}_i) - H(\mathbf{x}) = D_{KL}\left(p(\mathbf{x}) \| \prod_{i=1}^{d} p(\mathbf{x}_i)\right). \tag{4.7}$$

By non-negativity of TC, Equation 4.6 naturally forms variational lower bound $TC(\mathrm{x})$ to the CorEx objective, i.e., $TC(\mathrm{x}) \geq \mathcal{L}(\theta; \mathrm{x})$ for any θ. Equation 4.6 can be rewritten in terms of mutual information $I(\mathrm{x}; \mathrm{z}) = H(\mathrm{x}) - H(\mathrm{x} \mid \mathrm{z}) = H(\mathrm{z}) - H(\mathrm{z} \mid \mathrm{x})$. Further constraining the search space $p_\theta(\mathbf{z} \mid \mathbf{x})$ to have the factorized form $p_\theta(\mathbf{z} \mid \mathbf{x}) = \prod_{i=1}^{m} p_\theta(\mathbf{z}_i \mid \mathbf{x})$ and the mutual information terms can be bounded by approximating the conditional distributions $p_\theta(\mathbf{x}_i \mid \mathbf{z})$ and $p_\theta(\mathbf{z}_i \mid \mathbf{x})$ and parameterized by variational parameters α and ϕ with arbitrary distribution $r_\alpha(\mathbf{z})$. Finally, we can further rewrite and derive the lower bound for the objective function:

$$\begin{aligned}\mathcal{L}(\theta; \mathrm{x}) &= \sum_{i=1}^{d} I_\theta(\mathrm{x}_i : \mathrm{z}) - \sum_{i=1}^{m} I_\theta(\mathrm{z}_i : \mathrm{x}) \\ &\geq \left(\sum_{i=1}^{d} H(\mathrm{x}_i)\right) + E_{p_\theta(\mathrm{x},\mathrm{z})}\left(\log \underbrace{q_\phi(\mathrm{x} \mid \mathrm{z})}_{\text{decoder}}\right) \\ &\quad - D_{KL}(\underbrace{p_\theta(\mathrm{z} \mid \mathrm{x})}_{\text{encoder}} \| r_\alpha(\mathrm{z})).\end{aligned} \tag{4.8}$$

4.2.4 Total Correlation Variational Autoencoder (β–TCVAE)

Chen et al. [3] proposed disentanglement in their learned representations by adjusting the functional structure of the ELBO objective. The authors argued that each dimension of a disentangled representation should be able to represent a different factor of variation in the data and be changed independently of the other dimensions. β-TCVAE modifies the originally proposed ELBO objective by Higgins et al. [2] forcing the algorithm to learn

representations without explicitly making restrictions or reduction to the latent space. Recall the ELBO objective function (Equation 4.3) of β-VAE:

$$\mathcal{L}_{\beta VAE}(\theta, \phi) = \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x \mid z)\right] - \beta D_{\mathrm{KL}} \left(q_\phi(z \mid x) \| p(z)\right) \tag{4.9}$$

To introduce TC and disentanglement into the original β-VAE, Chen et al. decomposed the original KLD into **Index-Code MI, Total Correlation** and **Dimension-wise KL** terms. Furthermore, in the ELBO TC-Decomposition, each training samples are identified with a unique index **n** and a uniform random variable that refers to the aggregated posterior as $q(z) = \sum_{n=1}^{N} q(z \mid n)p(n)$ and can be denoted as:

$$\begin{aligned} \mathbb{E}_{p(n)}[\mathrm{KL}(q(z \mid n) \| p(z))] = \mathrm{KL}(q(z, n) \| q(z)p(n)) + \mathrm{KL}\left(q(z) \| \prod_i q\left(z_i\right)\right) \\ + \sum_i \mathrm{KL}\left(q\left(z_i\right) \| p\left(z_i\right)\right) \end{aligned} \tag{4.10}$$

Finally, with a set of latent variables z_i, with known factors v_k, the authors introduced a disentanglement measuring metric called mutual information gap (MIG) and defined in terms of empirical mutual information $I_n\left(z_i; v_k\right)$:

$$\frac{1}{K} \sum_{k=1}^{K} \frac{1}{H\left(v_k\right)} \left(I_n\left(z_{i(k)}; v_k\right) - \max_{i \neq j(k)} I_n\left(z_i; v_k\right)\right) \tag{4.11}$$

Here, $j^{(k)} = \mathrm{argmax}_j \, I_n\left(z_i; v_k\right)$ and K is the number of known factors under v_k.

4.3 Disentangled Conditional Variational Autoencoder (dCVAE)

Our approach builds on CorEx and models known sources of variation in the data, in a manner similar to Conditional Variational Autoencoder (CVAE) [60]. In what follows, we will represent this known source of variation using the variable C. In the experiment below, C is discrete and represents the class of each image. Modifying Equation 4.6 to incorporate C, we get

$$\mathcal{L}(\theta; x, c) = TC_\theta(x \mid c) - TC_\theta(x \mid z, c) - TC_\theta(z \mid c). \tag{4.12}$$

Recall that the first two terms measure the amount of correlation explained by z, and by maximizing it, we maximize the informativeness of the latent representation. The third term measures the correlation between the components of z, and by minimizing it, we maximize the disentanglement between the latent dimensions.

Using Mutual Information Theory [104], we can define the conditional differential entropy of $H(x)$ given c and interpret mutual information as a reduction in uncertainty after conditioning:

$$\begin{aligned} I(x; z \mid c) &= H(x \mid c) + H(z \mid c) - H(x, z \mid c) \\ I(x; z \mid c) &= H(x \mid c) - H(x \mid z, c) = H(z \mid c) - H(z \mid x, c). \end{aligned} \tag{4.13}$$

We can now rewrite Equation 4.12 using derived mutual information theory from Equation 4.13:

$$\mathcal{L}(\theta; x, c) = \sum_{i=1}^{n} I\left(x_i; z \mid c\right) - \sum_{i=1}^{m} I\left(z_i; x \mid c\right). \tag{4.14}$$

Now, consider the KLD between $p_\theta(\mathbf{x} \mid \mathbf{z}, c)$ and an approximating distribution $q_\phi(\mathbf{x} \mid \mathbf{z}, c)$. In terms of expectations with respect to the joint distribution $p_\theta(\mathbf{x}, \mathbf{z} \mid c)$, we can write:

$$-H(x \mid z, c) = E(\log p_\theta(x \mid z, c)) \geq E(\log q_\phi(x \mid z, c)). \tag{4.15}$$

Combing Equation 4.14 and 4.15 and assuming an approximating arbitrary distribution $r_\alpha(z_i \mid c)$ under variational parameter α and ϕ for $p_\theta(z_i \mid c)$, we obtain two inequalities:

$$I\left(x_i; z \mid c\right) = H\left(x_i \mid c\right) - H\left(x_i \mid z, c\right) \geq H\left(x_i \mid c\right) + E(\log q_\phi(x \mid z, c)), \tag{4.16}$$

$$I\left(z_i; x \mid c\right) = D_{KL}\left(p_\theta\left(z_i \mid x, c\right) \| r_\alpha\left(z_i \mid c\right)\right). \tag{4.17}$$

Combining these bounds with β, we finally derive a lower bound for the objective function for dCVAE:

$$\mathcal{L}(\theta; x, c) \geq \sum_{i=1}^{n} H\left(x_i \mid c\right) + E\left(\log q_\phi(x \mid z, c)\right) - \sum_{i=1}^{m} \beta D_{KL}\left(p_{(z_i \mid x, c)} \| r(z_i \mid c)\right). \tag{4.18}$$

Equation 4.18 illustrates the lower bound objective function of dCVAE where $q_\phi(\mathbf{x} \mid \mathbf{z}, c)$ is the generative model or decoder and $p_\theta(\mathbf{z}_i \mid \mathbf{x}, c)$ is the recognition model or encoder.

4.4 Experiments

In the experiments below, we compare our dCVAE method to five baseline methods: VAE, CVAE, β-VAE, Factor-VAE, and RFVAE. The first two methods were selected as well-known baselines that do not explicitly enforce disentanglement; on the other hand, the latter three methods seek to achieve a disentangled representation of the data.

4.4.1 Datasets

We evaluate dCVAE and other baseline models on the following four datasets. MNIST [72], Fashion-MNIST (FMNIST) [95] are considered as the benchmark dataset whether-as the KMNIST [105] and EMNIST [106] are used for testing accuracy on a real-world dataset to assess overall performance. A more detailed description of these datasets follows:

- **MNIST and FMNIST:** Firstly, we apply all models to two benchmark datasets, MNIST and FMNIST, for a fair comparison with other baseline methods. We used all 10 classes with 60000 and 10000 training and testing samples for both datasets with $28 \times 28 \times 1$ pixels channel.
- **KMNIST:** Secondly, we applied the same training process to another complex real-world dataset, Kuzushiji-MNIST or KMNIST. KMNIST is a drop-in replacement for the MNIST dataset, a Japanese cursive writing style. KMNIST contains similar 10 classes with 60000 and 10000 training and testing samples with $28 \times 28 \times 1$ pixels channel.
- **EMNIST:** Finally, all models are tested on the Extending MNIST or EMNIST Dataset to further assess the performance. EMNIST has extended 62 classes (digit 0-9, letters uppercase A-Z and lowercase a-z) with 700000 and 80000 training and testing samples with $28 \times 28 \times 1$ pixels channels. As a result, this dataset posses more challenges for the methods while conducting the downstream tasks. The EMNIST dataset was processed from NIST Special Database 19 [107] and contains handwritten digits and characters collected from over 500 writers.

4.4.2 Reconstruction error and Anomaly Score

Leveraging methods for the discriminator as the anomaly score and drawing separation between normal and anomalous data is challenging for the divergent architectures of autoencoders. Depending on the task the architecture is trained for, the discriminator varies greatly. In general, the UAD methods utilize reconstruction error [108], distribution-based error [109], and density-based error [110] scores to distinguish normal and anomalous data. Formally, for each input x, a test input $\widehat{x_l}$ is considered to be anomalous if reconstruction error or Anomaly score($\mathcal{A}$) is greater than the minimum threshold value and denoted as follows:

$$\mathcal{A}(\hat{x}) = \|x - \mathrm{D}(\mathrm{G}(\hat{x}))\|_2. \tag{4.19}$$

4.4.3 Performance Metrics

One of the challenges of measuring the performance of disentanglement is to apply appropriate metrics based on the nature of the dataset, not of latent factors or dimensions in the latent space. Therefore, considering the different model architectures and datasets, we first measure the performance using Numerical AUC Score, reconstruction error ($\mathcal{A}$), and negative ELBO score ($\mathcal{E}$). These metrics provide a quantifiable method of accuracy, while also measuring the disentanglement among the latent factors.

We also measure performance qualitatively by visualizing the latent space and the 2D-manifold. Both allow us to visualize the orthogonality between latent features and demonstrate the accuracy of the models to handle reduced latent variables and the ability to reconstruct samples.

4.4.4 Model configuration

A fixed set of hyper-parameters are chosen to formulate a similar platform for all models and identify the computational cost and reproducibility of the models. Although baseline models that we chose, β-VAE, FactorVAE, RFVAE are highly sensitive to hyper-parameters tuning, the hyper-parameters throughout the experiment are kept consistent to observe how the models perform under similar values. A minimal 50 epochs are used to train the datasets. For MNIST, FMNIST, and KMNIST the batch size is kept to 64, with primary and secondary learning rates as $\alpha = 10^{-5}$ and $\alpha = 10^{-3}$ respectively.

However, for the EMNIST dataset, the batch size increased to 128, and learning rates as $\alpha = 10^{-6}$ and $\alpha = 10^{-5}$.

4.5 Results and Discussion

In this section, we evaluate the results of dCVAE and other baseline methods on the downstream task of anomaly detection. A considerable volume of results was produced from our exhaustive evaluation. However, accounting for limitations of space here, we elected to focus on the results from EMNIST and KMNIST datasets in the main text. The remaining results (MNIST and FMNIST) are presented as Supplementary Material.

We show the results of our evaluation in three stages: firstly, using sample reconstruction and the negative ELBO score ($\mathcal{E}$) with reconstruction error $\mathcal{A}$, we evaluate and compare the disentanglement ability of dCVAE with baseline architectures. Secondly, we use the UMAP algorithm [111] to reduce dimensions and visualize both latent representation, as well as interpolation of the 2D-manifold to distinguish the TC by comparing information loss and effects of modeling known sources of variation. Finally, we present AUC scores and training time to summarize the overall accuracy of the experimented methods.

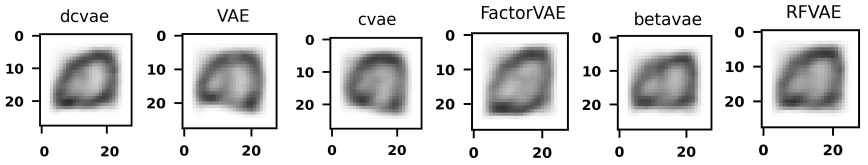
We evaluate the quality of disentanglement by considering explicit separation of $\mathcal{A}$ between normal and anomalous data and minimization of $\mathcal{E}$. A better disentanglement is achieved when:

(a) A higher reconstruction error $\mathcal{A}$ for anomalous sample and lower reconstruction error $\mathcal{A}$ for normal sample is obtained and

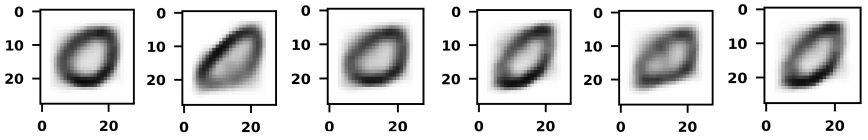
(b) $\mathcal{E}$ is minimized by enforcing regularization that either minimizes the negative ELBO decomposition $D_{KL}\left(p_{(z_i|x,c)\|r(z_i|c)}\right)$ or regularizes the approximate posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$.

A clear boundary in terms of learning efficient disentanglement between dCVAE and baseline methods can be observed from both EMNIST (Figure 4.1) and KMNIST (Figure 4.2) reconstruction. The first row corresponds to anomalous reconstruction and the second row shows normal sample reconstruction. Both $\mathcal{E}$ and $\mathcal{A}$ score suggests that dCVAE captures more independent factors and identifies anomalous and normal samples efficiently. This observation strongly justifies one of our primary claims, namely that dCVAE incorporates the disentanglement learning through enforcing TC and restrict independent latent variables to prioritize the minimization of the divergence. The other

disentanglement methods presented here either only emphasize TC (indicated by the dependence between random variables) or introduce β (weighing the prior enforcement term), which limits the ability to learn randomness in a case when the hyperparameters are not tuned for certain dimensions.



(A) $\mathcal{E}$ = -235 $\mathcal{A}$ = 0.97 (B) $\mathcal{E}$ = -248 $\mathcal{A}$ = 0.90 (C) $\mathcal{E}$ = -245 $\mathcal{A}$ = 0.91 (D) $\mathcal{E}$ = -246 $\mathcal{A}$ = 0.91 (E) $\mathcal{E}$ = -244 $\mathcal{A}$ = 0.92 (F) $\mathcal{E}$ = -246 $\mathcal{A}$ = 0.92



(G) $\mathcal{E}$ = -171 $\mathcal{A}$ = 0.33 (H) $\mathcal{E}$ = -199 $\mathcal{A}$ = 0.48 (I) $\mathcal{E}$ = -180 $\mathcal{A}$ = 0.53 (J) $\mathcal{E}$ = -185 $\mathcal{A}$ = 0.52 (K) $\mathcal{E}$ = -195 $\mathcal{A}$ = 0.57 (L) $\mathcal{E}$ = -190 $\mathcal{A}$ = 0.55

FIGURE 4.1: Reconstruction for digit zero (0) and the capital letter O. Here, $\mathcal{E}$ refers to Negative ELBO score and $\mathcal{A}$ is the reconstruction error or anomaly score. Only dCVAE and FactorVAE show steady improvement for both types of reconstruction. All the other methods misclassify the samples. Moreover, we can observe higher reconstruction error and ELBO scores compared to MNIST (Figure A.1) and FMNIST (Figure A.2).

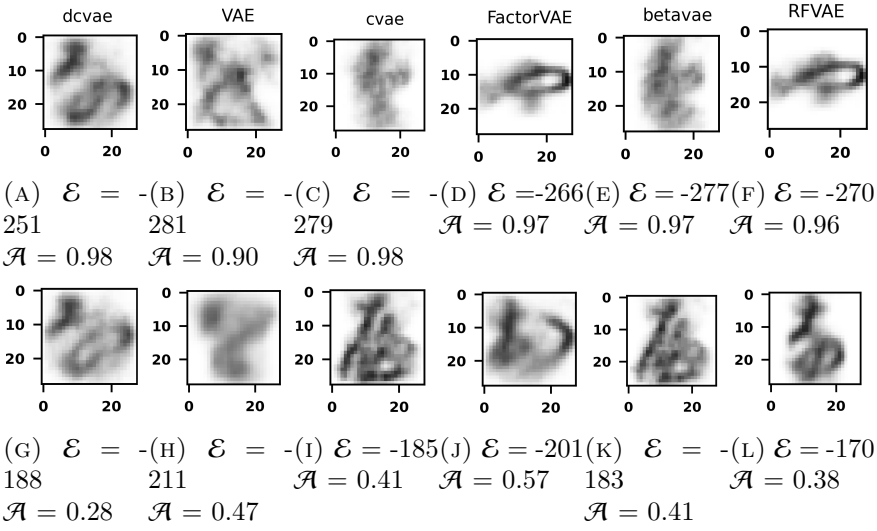
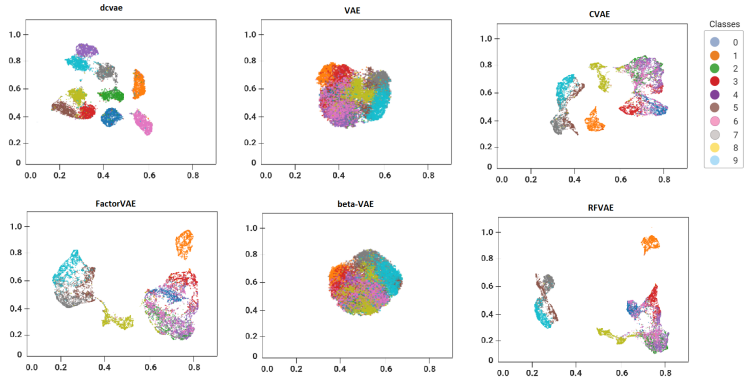
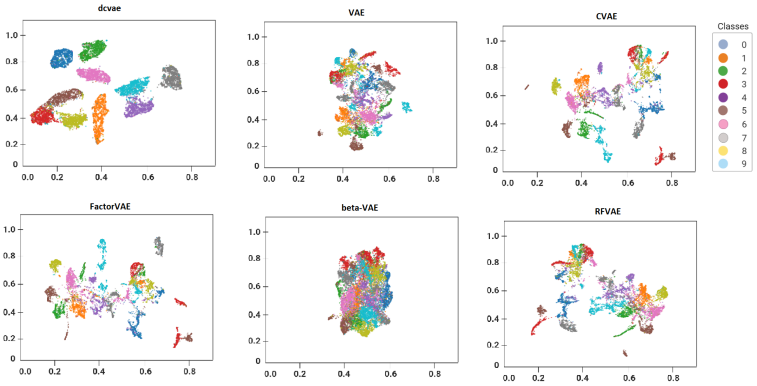


FIGURE 4.2: In KMNIST dataset, without dCVAE, all other methods fail to classify both anomalous and normal samples. Reconstruction scores suggest FactorVAE, VAE almost fail to distinguish normal and anomalous observations. Since the stroke of the samples are similar in this dataset, methods that only emphasize disentanglement or empirical approximation lose more information in latent variable resulting in false anomaly detection.

The second observation is drawn using latent representation (Figure 4.3) and 2D-manifold embeddings (Figure 4.4 and 4.5). Through this experiment, we observe the effect of modeling using a known source of variation (i.e. introducing conditional variable C into the objective function) and minimizing information loss through multivariate mutual information theory (i.e. decomposition of TC). We can observe clear similarities between KLD loss and modeling with known score of variance in a reduced latent space. Due to enforced divergence loss, the plot of VAE and β-VAE are noticeably different from other architectures. Feature space is more compact for VAE, β-VAE, and we can see the cluster of the different classes are not well separated. However, conditioning the generative function (encoder) of CVAE and dCVAE provides the leverage to construct higher feature space and retain more accurate information in 2D-manifold (EMNIST, Figure 4.4; and KMNIST, Figure 4.5). Furthermore, TC reduces the correlation among disentanglement degrees when a specific feature is learned (shape, strokes, color, boundaries). Such classes can be observed to cluster together and the other gets scattered with higher feature space (Figure 4.3). Compared to other methods, it is evident that dCVAE maintain consistent latent space and create separate clusters more accurately. This indicates that more disentangled variables are captured, and they retain more information through conditioning the generative model by minimizing the ELBO $D_{KL}\left(p_{(z_i|x,c)}\|r(z_i|c)\right)$.

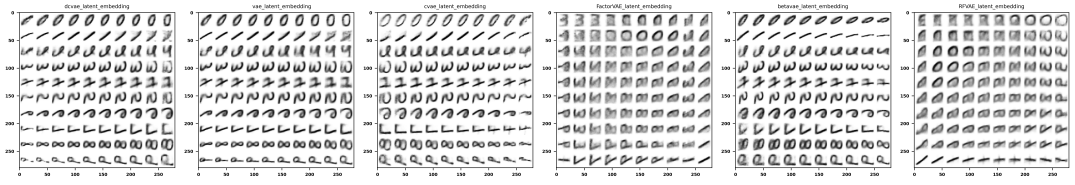


(A) EMNIST



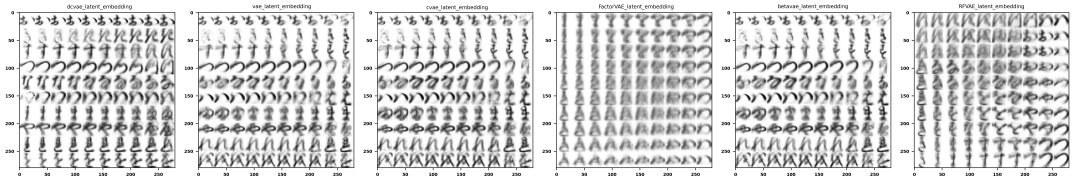
(B) KMNIST

FIGURE 4.3: Latent Representation of EMNIST and KMNIST



(A) dCVAE (B) VAE (C) CVAE (D) FactorVAE (E) β-VAE (F) RFVAE

FIGURE 4.4: Manifold Embeddings (EMNIST)



(A) dCVAE (B) VAE (C) CVAE (D) FactorVAE (E) β-VAE (F) RFVAE

FIGURE 4.5: Manifold Embeddings (KMNIST)

Finally, Table 4.1 illustrates the results of model evaluation through AUC score and training time. dCVAE outperforms other methods in terms of AUC score. However, for larger divergent datasets like KMNIST and EMNIST, VAE shows lower training time compared to dCVAE. Since VAE only optimizes the negative log-likelihood, reconstruction loss and prior enforcement term, the training takes fewer latent variables to regularize, resulting in less training time. Nevertheless, compared to methods that incorporate TC (e.g. FactorVAE and RFVAE) or a constraint on the posterior (β-VAE), our proposed dCVAE scales to all larger datasets with higher classification accuracy.

TABLE 4.1: Evaluation metrics score

Model	MNIST		FMNIST		EMNIST		KMNIST	
	AUC	Training Time (min)	AUC	Training Time (min)	AUC	Training Time (min)	AUC	Training Time (min)
dCVAE	**88.31**	37	**88.63**	44	**78.98**	102	**61.02**	95
VAE	88.21	37	84.12	39	67.23	92	51.13	78
CVAE	87.57	43	83.31	48	66.01	117	42.35	104
FactorVAE	87.11	53	82.78	50	62.91	138	49.23	117
β-VAE	85.31	51	82.31	53	65.12	123	50.01	119
RFVAE	85.31	55	81.11	57	55.03	130	49.51	132

The only trade-offs in our proposed method seem to occur when minimizing the negative ELBO loss. In certain conditions, dCVAE reaches a lower reconstruction loss (anomalous sample) yet minimizes the negative ELBO score (Figure 4.3, 4.4). In general, negative ELBO loss should illustrate symmetrical change with reconstruction error. Such inconsistency could lead to a significant drop in the classification accuracy, thus leading to a false anomaly detection result.

4.6 Conclusion

In this research, we present a novel generative variational model dCVAE, to improve the unsupervised anomaly detection task through disentanglement learning, TC loss, and minimizing trade-offs between reconstruction loss and reconstruction quality. Introducing a conditional variable to mitigate the loss of information effectively captures more disentangled features and produces more accurate reconstructions. Such architecture could be used in a wider range of applications, including generating controlled image synthesis, efficient molecular design and generation, source separation for bio-signals and images, and conditional text generation. Future research direction includes investigating in the gap between the posterior and the prior distribution, resolving the trade-offs between loss function and reconstruction, and inspect dCVAE using different disentanglement metrics.

Reproducibility statement

In this research, we carefully considered reproducibility in designing and conducting all experiments. In our supplemental texts, we have attached our source code. The experiments are designed independently to make the results reproducible. Image reconstruction and generation, 2D-Manifold embeddings, training time, and ELBO score calculation are performed separately from other downstream tasks like classification accuracy, reconstruction error, and latent representation. Furthermore, we used both TensorFlow and PyTorch frameworks to remove package dependencies. To remove the library dependencies and installation issues, virtual environment and package requirement files are also added. Finally, to make the results more accessible, we also provided randomly generated images with supplementary texts.

Acknowledgments

We would like to acknowledge support from the NSERC CREATE grant on Visual and Automated Disease Analytics (VADA) program. MT (RGPIN-2021-04073) gratefully acknowledges funding via a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada (NSERC).

Chapter 5

Conclusion

5.1 Summary

In this thesis, I presented two manuscripts corresponding to chapters 3 and 4. Firstly, in Chapter 2, I defined anomaly and anomaly detection, and then I explained the primary background of types of anomalies, anomaly detection methods, and framework related to unsupervised methods. Afterward, I presented the comparison between the autoencoder and other unsupervised methods to illustrate the autoencoder's appeal as the primary technique for this thesis. Finally, I presented a few closely related VAE techniques, namely CVAE, β-VAE, FactorVAE, RFVAE, FVAE, InfoVAE and CorEX, to present the association between utilizing multivariate information theory for efficient disentangled learning and improving the ability of anomaly detection.

In Chapter 3, I reviewed the architectures of the autoencoder for the task of UAD. As discussed in Chapter 2, a key feature for efficient anomaly detection is learning the meta-priors. In particular, VAEs and their extensions addressing meta-priors have shown great potential in detecting anomalies in images dataset. However, two essential parts remained unanswered: how to combine VAE architecture to balance information loss and retain the quality of generated samples. Then, identify disentangled factors to regularize VAEs loss function to efficiently retain information loss. Separated in three distinct categories, I experimented with 11 autoencoder architectures to identify such capabilities and leverage the connection between VAE architectures with meta-priors. Through experiments and extensive evaluations, it became evident that posterior-variant autoencoders(e.g. CVAE, β-VAE, VQ-VAE) are more efficient than VAE at AD tasks. Observations from

latent visualization and sample generations show that posterior variants utilizes the autoregressive prior to pair representation. For Example β-VAE emphasizes locating disentangled latent factors, CVAE conditions the generative models, CorEx introduces TC for minimizing information loss. I also showed that structured prior distribution is crucial for learning data representation and improving the blurry reconstruction of VAEs. Finally, I pointed out the advantages and limitations of current VAE extensions and propose future research directions for VAE architectures.

Finally, in Chapter 4, I showed how the idea of the literature review and the manuscript can be combined to improve unsupervised anomaly detection using disentangled learning and minimize information loss for efficient sample reconstruction. Specifically, I studied the trade-off between reconstruction loss and reconstruction quality. The backbone of VAE was combined with conditional VAE, which conditions on known sources of variation, and mutual information theory, which learns disentangled data representations while minimizing the loss of information. As a consequence, we expected good reconstruction capabilities and superior performance for AD tasks. Using four different datasets, I showed using anomaly score and ROC-AUC metrics that this was indeed the case: the reconstruction quality heavily depends on retaining information by minimizing the loss of the objective function. Additionally, I emphasized model configuration while training the models to attain training times for observing model complexity and reproducing interpretable and constructive results. A clear improvement in sample reconstruction, optimizing TC, and overall improving the UAD can be observed from the proposed dCVAE framework. However, it is also observed that there is a trade-off between the negative ELBO score optimization and the anomaly score. In general, the dCVAE shows excellent potential in a broad range of applications, including generating controlled image synthesis, efficient molecular design and generation, source separation for bio-signals and images, and conditional text generation.

5.2 Limitations

In Chapter 2, I demonstrated the advantage and limitations of different VAE architectures using the reconstruction, training time, AUC score, and latent space visualization. However, selecting the VAE architectures is not robust. I chose the principle architectures, although recent literature reviews show such architectures and improved and new frameworks are already proposed. Model Selection and discovering highlights and challenges can be improved by selecting diverse model categories of VAE architectures such

as: Adversarial Conditional AE [112], Stochastic Recurrent Networks [113], Sparse Gaussian Process VAE [114], and Hypergraph VAE [115]. Another area of improvement is using variations of accuracy metrics and inspecting the latent variables using numerical estimations.

In Chapter 4, introducing disentangled metrics for dCVAE and improving baseline comparisons are still due in this study. Compared to other meta-priors, disentangled representation depends on the factors and variants of the data. It can produce wrong premises about the data if they are measured using inadequate metrics. Therefore, further study is required to find the right accuracy metrics for dCVAE. Furthermore, previously proposed disentangled learning models, such as β-VAE and FactorVAE, deliberately depend on tuning parameters according to datasets and model configuration. In my proposed framework, the hyperparameters are kept uniform for all methods, that might result in lower accuracy for datasets like EMNIST and KMNIST. However, Tuning parameters can also affect the overall baseline comparison criteria, such as accuracy metrics, classification accuracy, and training time. For this reason, I excluded the ablation study of hyperparameters from consideration for this thesis.

5.3 Future work

The main chapters of this thesis, Chapters 3 and 4, highlight three potential directions for future research. First, reviewing autoencoder and dCVAE frameworks are applied in a uniform set of datasets: MNIST, Fashion-MNIST, EMNIST, and KMNIST. The latter two datasets contain highly diverse real-world data samples and larger classes and samples. Despite having such miscellaneous samples, dCVAE shows great accuracy over other methods. A potential avenue for dCVAE can be applying this framework for structural anomaly detection and detecting abnormalities in molecular generations where segmenting and classifying images in a supervised manner is critical. In such a context where the samples are high-dimensional and contain correlated features, learning disentangled factors and minimizing loss of information can significantly improve detecting abnormalities.

The second and third avenues for dCVAE are improving pixel-wise class detection and introducing contextual anomaly detection [116]. In an unsupervised manner, defining anomalous data boundaries highly depends on the course of the problem and changes over time. Current literature depends on a non-trivial attention-based mechanism. The dCVAE framework can be an exciting experiment to apply in diverse examples like molecular controls and generating chemical substances, particularly in the genome-sequence

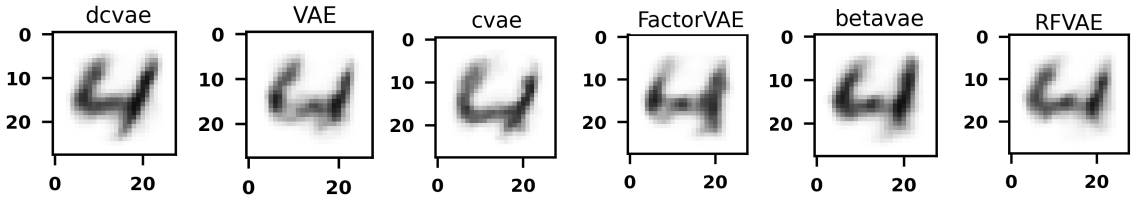
research domain. I am interested in applying dCVAE in discrete datasets (textual, video) and acquiring experimental observations to improve this framework.

Appendix A

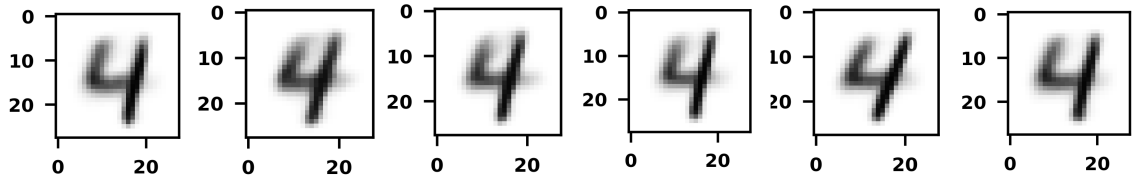
Appendix to Manuscript 1

Since we couldn't accommodate all results in our main paper, in this section we present results produced from MNIST and FMNIST datasets. The results are categorized into three sections: Reconstructions (A.1), Latent Representation (A.2), and 2D-Manifold embeddings (A.3).

A.1 Reconstruction

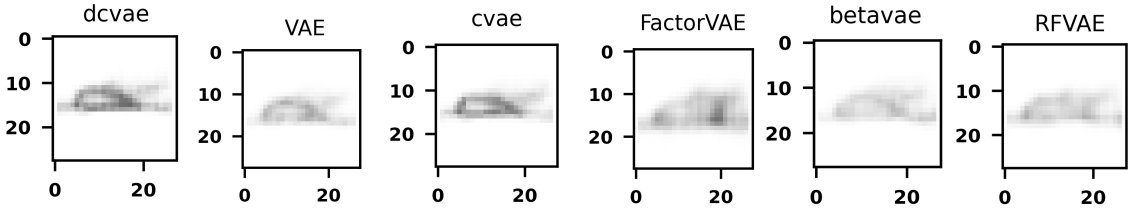


(A) $\mathcal{E}$ = -152, $\mathcal{A}$ = 0.88 (B) $\mathcal{E}$ = -181, $\mathcal{A}$ = 0.71 (C) $\mathcal{E}$ = -170, $\mathcal{A}$ = 0.75 (D) $\mathcal{E}$ = -166, $\mathcal{A}$ = 0.78 (E) $\mathcal{E}$ = -176, $\mathcal{A}$ = 0.76 (F) $\mathcal{E}$ = -178, $\mathcal{A}$ = 0.81

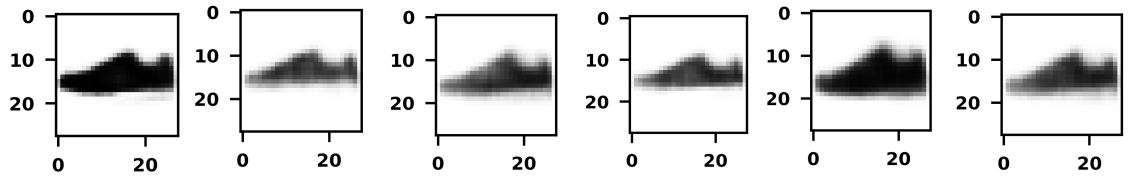


(G) $\mathcal{E}$ = -106, $\mathcal{A}$ = 0.23 (H) $\mathcal{E}$ = -140, $\mathcal{A}$ = 0.48 (I) $\mathcal{E}$ = -136, $\mathcal{A}$ = 0.32 (J) $\mathcal{E}$ = -132, $\mathcal{A}$ = 0.32 (K) $\mathcal{E}$ = -126, $\mathcal{A}$ = 0.30 (L) $\mathcal{E}$ -131, $\mathcal{A}$ = 0.31

FIGURE A.1: The reconstruction from the MNIST dataset shows similar negative ELBO and reconstruction error ($\mathcal{A}$) values for CVAE, β-VAE, and RFVAE. our proposed model dCVAE performs best in terms of both reconstructing anomalous observation (first row) and normal observation (second row). We can observe a trade-off in FactorVAE with respect to β-VAE and RFVAE. FactorVAE performs better in reconstructing the anomalous observation whether as the β-VAE shows good performance in normal observations.



(A) $\mathcal{E}$ = -187 $\mathcal{A}$ = 0.94 (B) $\mathcal{E}$ = -198 $\mathcal{A}$ = 0.79 (C) $\mathcal{E}$ = -192 $\mathcal{A}$ = 0.82 (D) $\mathcal{E}$ = -221 $\mathcal{A}$ = 0.90 (E) $\mathcal{E}$ = -231 $\mathcal{A}$ = 0.85 (F) $\mathcal{E}$ = -229 $\mathcal{A}$ = 0.92



(G) $\mathcal{E}$ = -151 $\mathcal{A}$ = 0.12 (H) $\mathcal{E}$ = -170 $\mathcal{A}$ = 0.37 (I) $\mathcal{E}$ = -162 $\mathcal{A}$ = 0.40 (J) $\mathcal{E}$ = -165 $\mathcal{A}$ = 0.41 (K) $\mathcal{E}$ = -155 $\mathcal{A}$ = 0.38 (L) $\mathcal{E}$ = -162 $\mathcal{A}$ = 0.42

FIGURE A.2: Similar to the MNIST dataset, the FMNIST illustrates similar trade-offs among FactorVAE, RFVAE, and β-VAE. However, for some samples, β-VAE mis-classifies the closely matched classes. dCVAE constrains the blurry reconstruction by enforcing conditions in the prior.

A.2 Latent Space Visualization

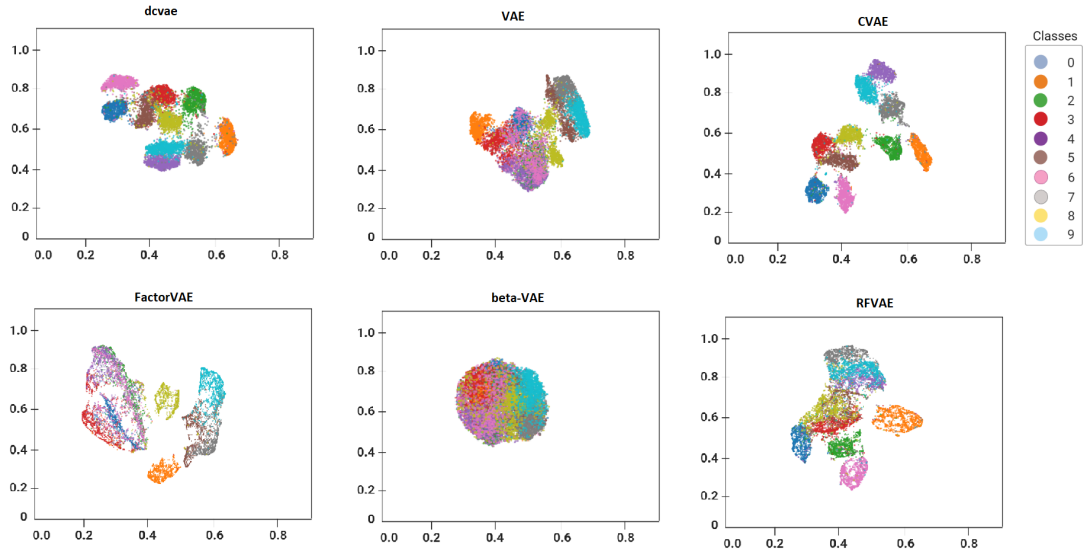


FIGURE A.3: Latent Space Representation (MNIST)

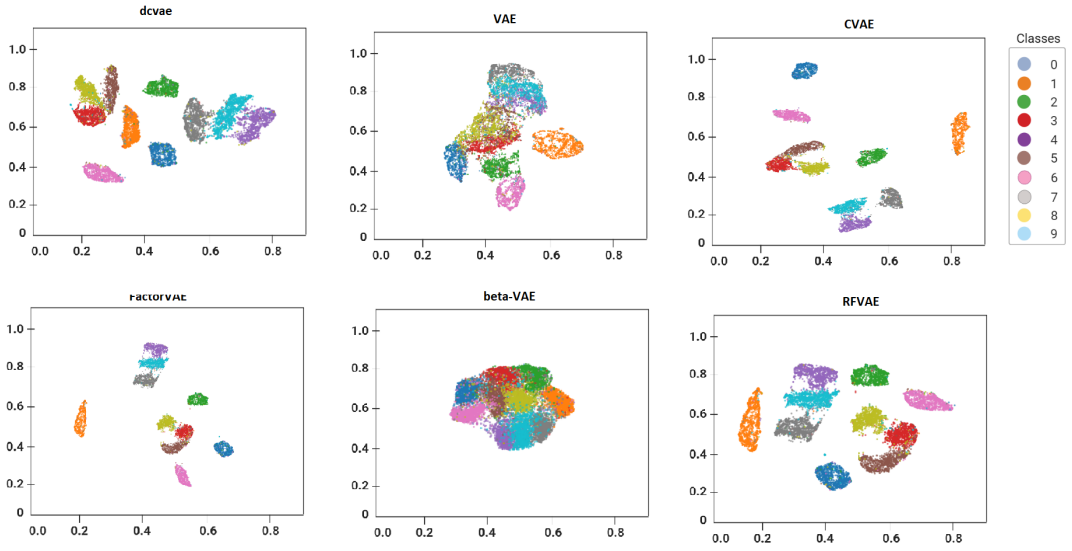
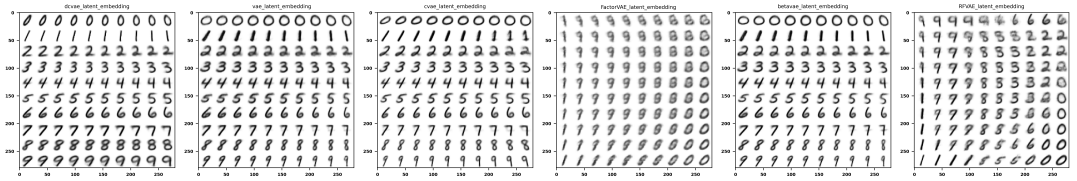


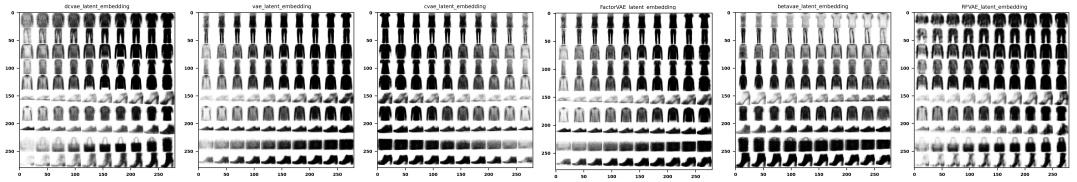
FIGURE A.4: Latent Space Representation (FMNIST)

A.3 Latent Manifold



(A) dCVAE (B) VAE (C) CVAE (D) FactorVAE (E) β-VAE (F) RFVAE

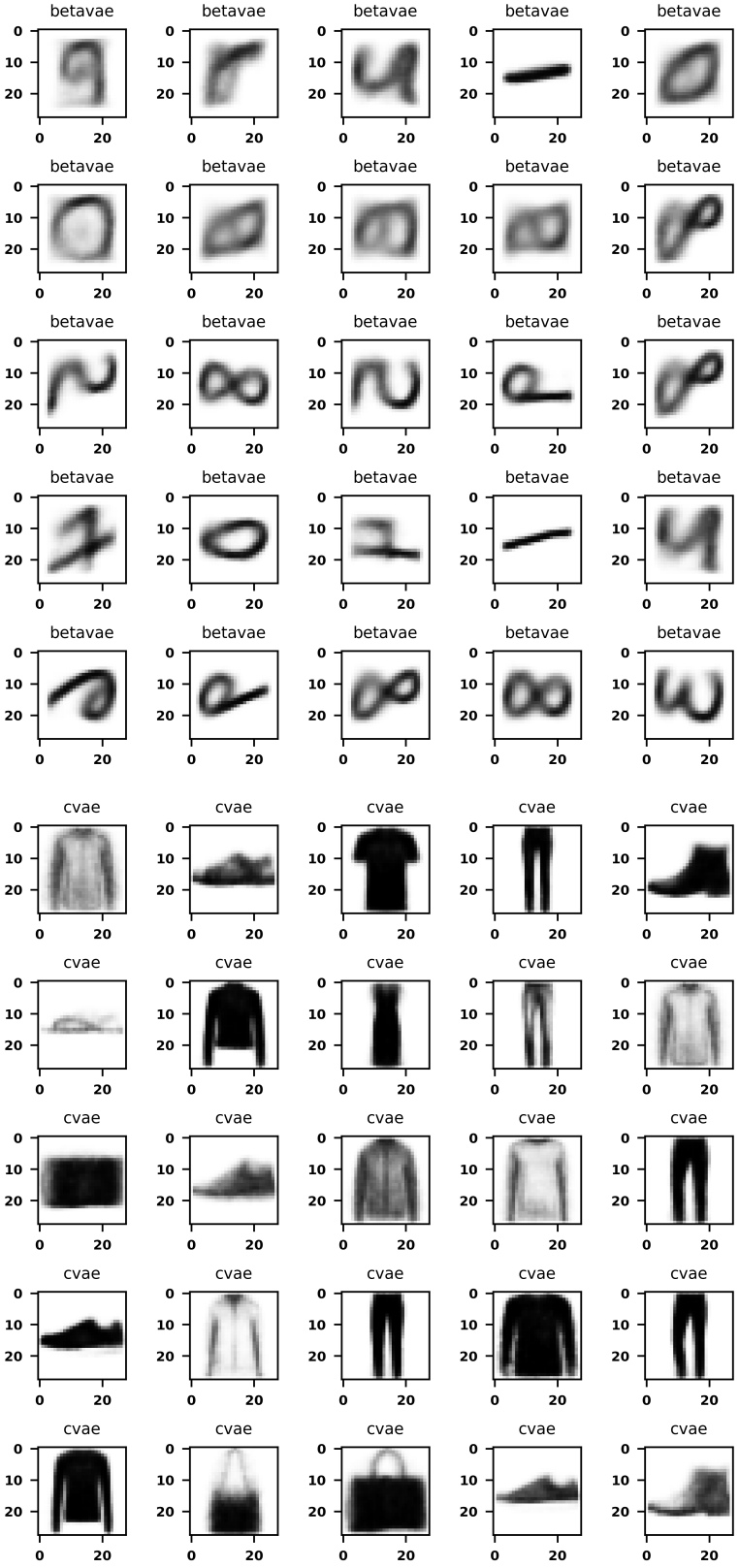
FIGURE A.5: Latent Embeddings (MNIST)

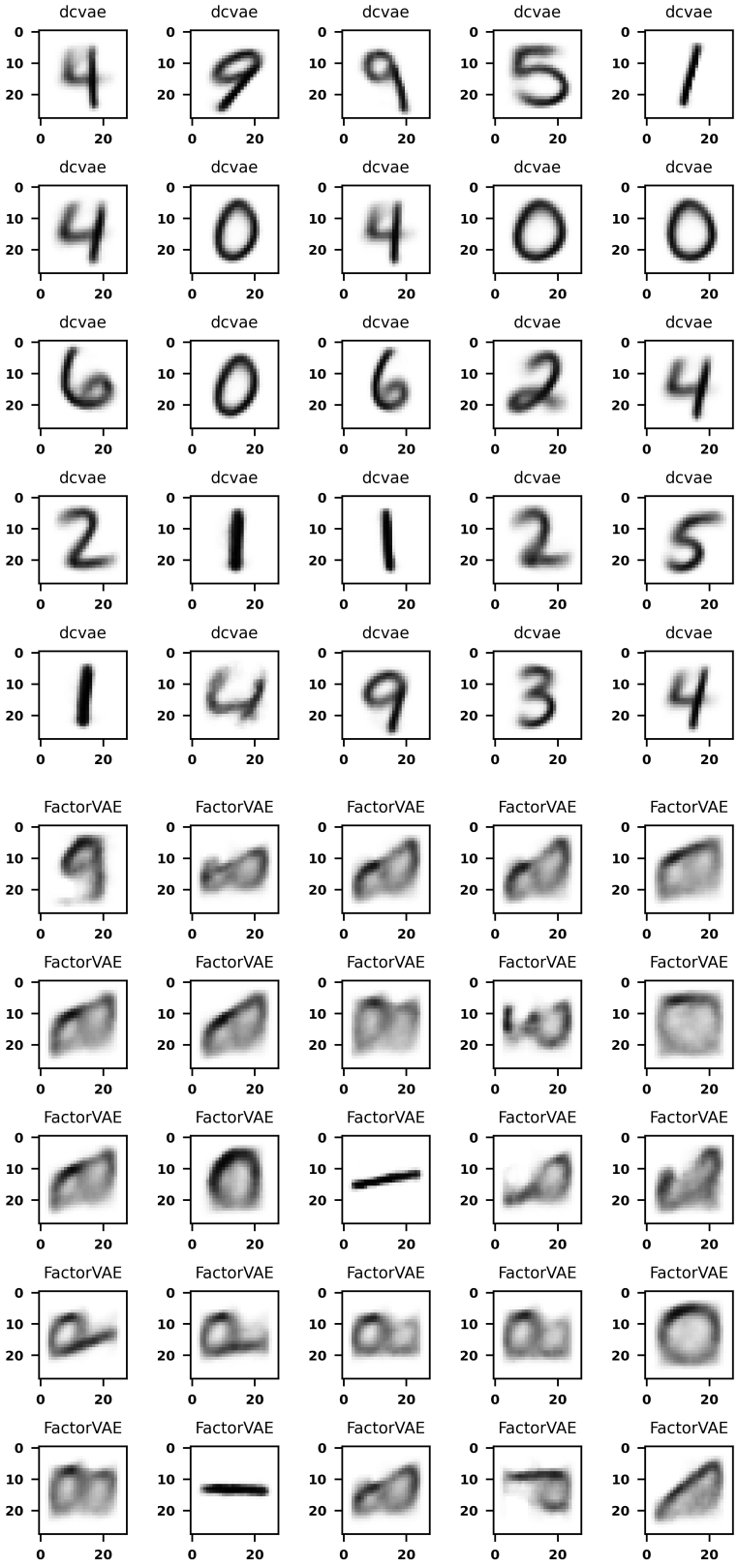


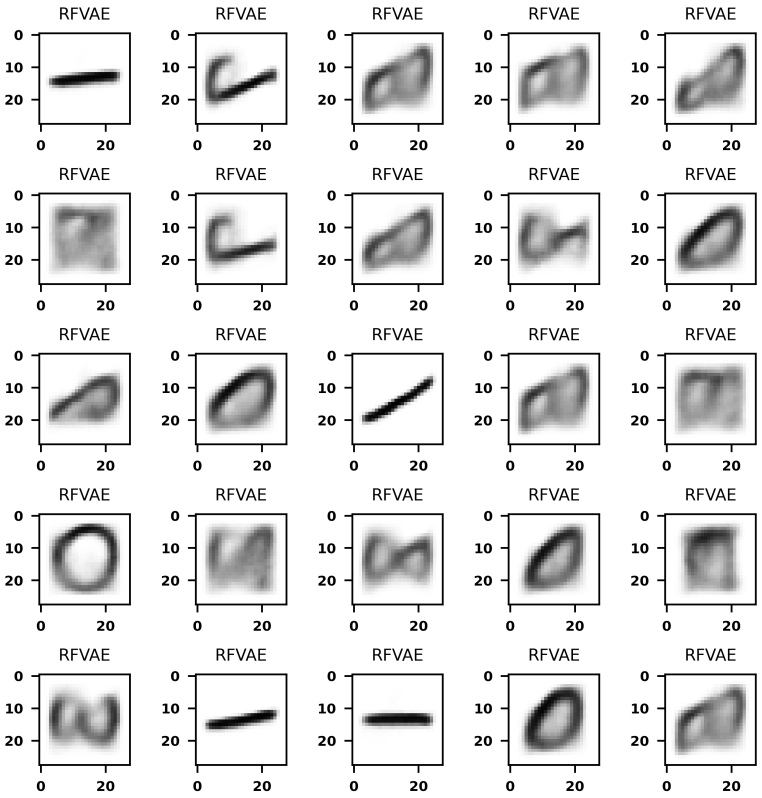
(A) dCVAE (B) VAE (C) CVAE (D) FactorVAE (E) β-VAE (F) RFVAE

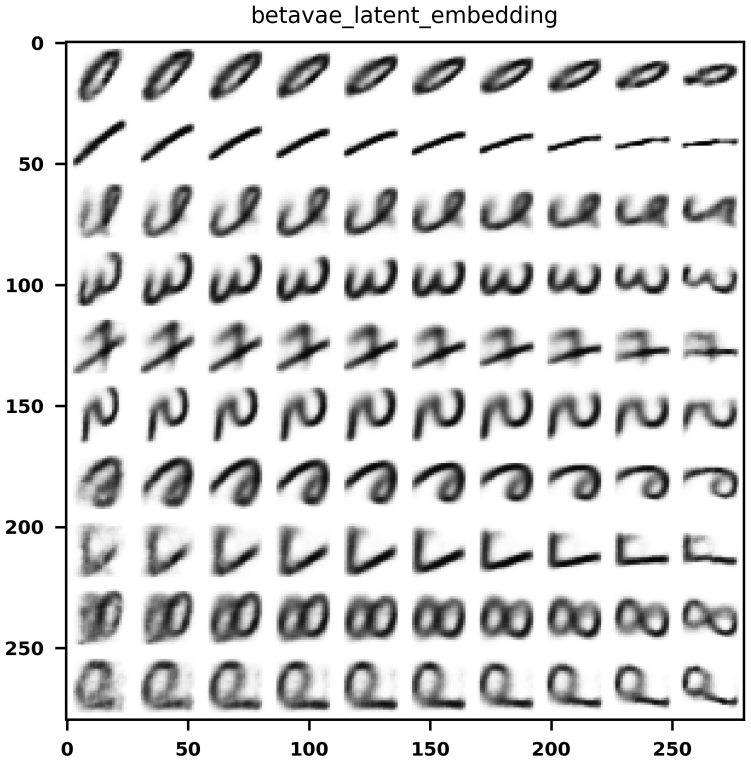
FIGURE A.6: Latent Embeddings (FMNIST)

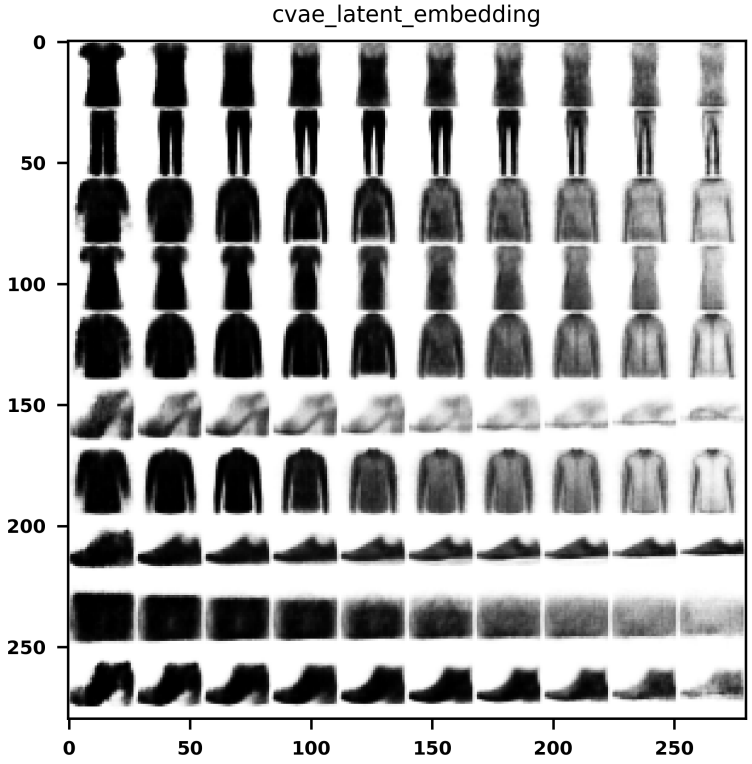
A.4 Random Generation

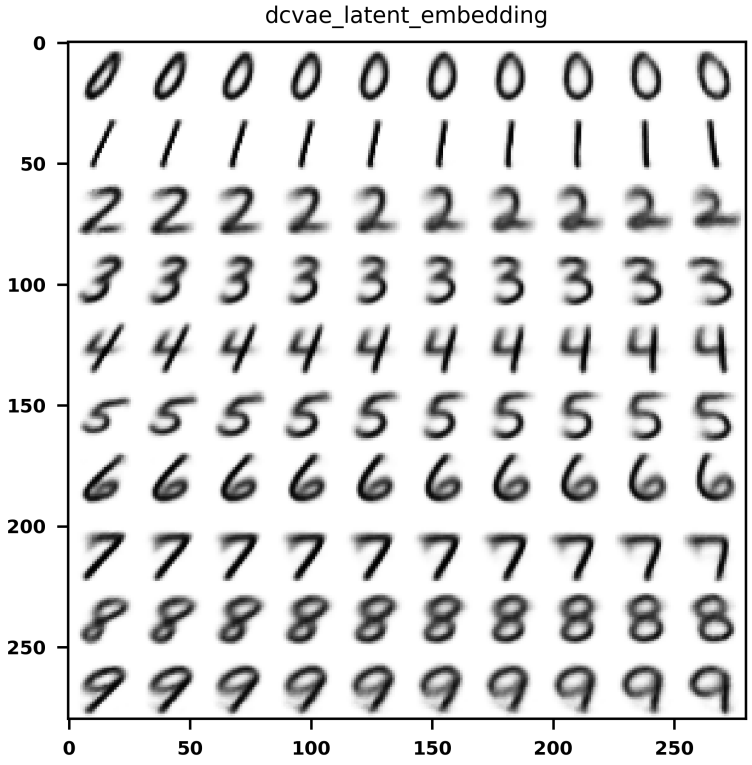


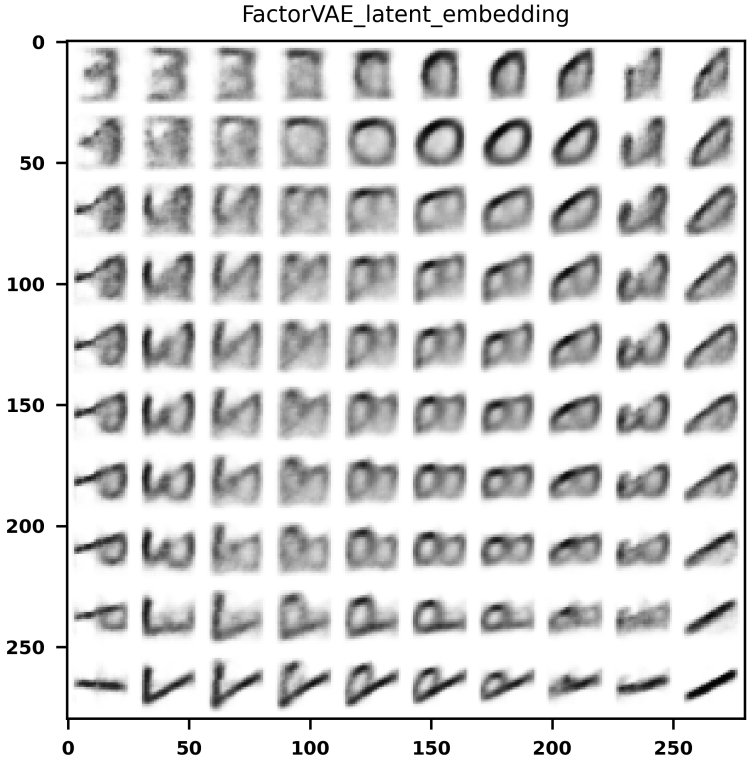

dcvae
FactorVAE

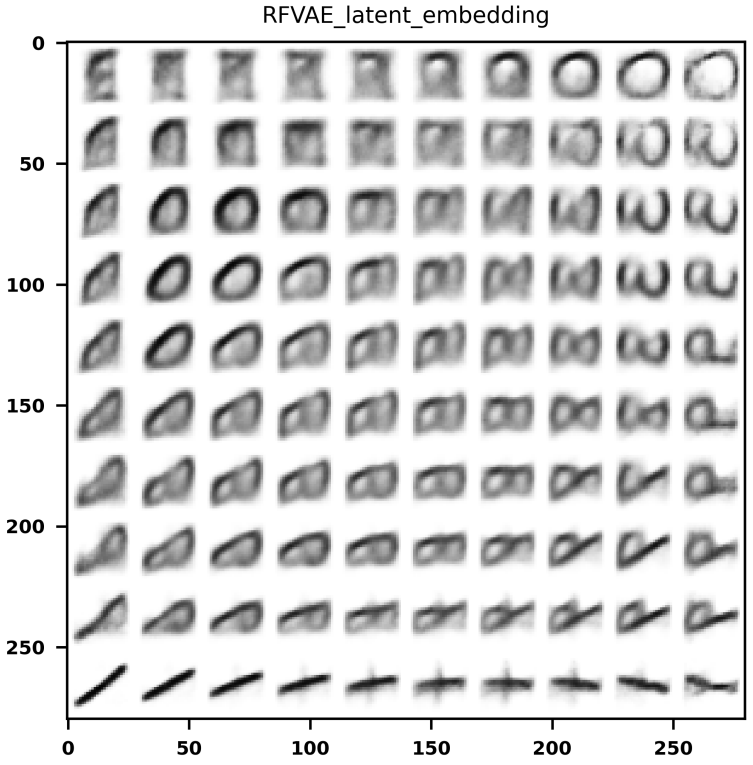

RFVAE


betavae_latent_embedding
0
50
100
150
200
250


cvae_latent_embedding
0
50
100
150
200
250
0
50
100
150
200
250


dcvae_latent_embedding
0
50
100
150
200
250
0
50
100
150
200
250


FactorVAE_latent_embedding
0
50
100
150
200
250
0
50
100
150
200
250


RFVAE_latent_embedding
0
50
100
150
200
250
0
50
100
150
200
250

Bibliography

[1] Leman Akoglu, Mary McGlohon, and Christos Faloutsos. Anomaly detection in large graphs. In *In CMU-CS-09-173 Technical Report*. Citeseer, 2009. viii, 8

[2] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-VAE: Learning basic visual concepts with a constrained variational framework. In *ICLR 2017 : 5th International Conference on Learning Representations*, 2017. 2, 15, 16, 25, 30, 63, 64, 66

[3] Ricky TQ Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in VAEs. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 2615–2625, 2018. 2, 16, 49, 63, 64, 66

[4] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018. 2, 15, 16, 63, 65

[5] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*, 2017. 2, 15, 16, 49, 63

[6] Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. The variational fair autoencoder. *arXiv preprint arXiv:1511.00830*, 2015. 2, 16, 49

[7] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. *arXiv preprint arXiv:1711.00848*, 2017. 2, 49

[8] Xiao Liu, Pedro Sanchez, Spyridon Thermos, Alison Q. O'Neil, and Sotirios A. Tsaftaris. Learning disentangled representations in the imaging domain. *Medical Image Analysis*, 80:102516, 2022. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2022.102516. URL `https://www.sciencedirect.com/science/article/pii/S1361841522001633`. 3

[9] J Gladitz. Barnett, v. & lewis, t.: Outliers in statistical data, john wiley & sons, chichester–new york–brisbane–toronto–singapore, 1984, xiv, 463 s., 26 abb.,£ 29.95, isbn 0471905070, 1988. 6

[10] Richard A Johnson, Dean W Wichern, et al. Applied multivariate statistical analysis. *New Jersey*, 405, 1992. 6

[11] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009. 7, 9, 10, 11, 22, 30

[12] Hongchao Song, Zhuqing Jiang, Aidong Men, and Bo Yang. A hybrid semi-supervised anomaly detection model for high-dimensional data. *Computational intelligence and neuroscience*, 2017, 2017. 7, 22

[13] Arman Mohammadi Gonbadi, Seyed Hasan Tabatabaei, and Emmanuel John M Carranza. Supervised geochemical anomaly detection by pattern recognition. *Journal of Geochemical Exploration*, 157:81–91, 2015. 9

[14] Sarah Erfani, Mahsa Baktashmotlagh, Masud Moshtaghi, Xuan Nguyen, Christopher Leckie, James Bailey, and Rao Kotagiri. Robust domain generalisation by enforcing distribution invariance. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*, pages 1455–1461. AAAI Press, 2016. 9

[15] Vilen Jumutc and Johan AK Suykens. Multi-class supervised novelty detection. *IEEE transactions on pattern analysis and machine intelligence*, 36(12):2510–2523, 2014. 9

[16] Sangwook Kim, Yonghwa Choi, and Minho Lee. Deep learning with support vector data description. *Neurocomputing*, 165:111–117, 2015. 9

[17] Magnus Almgren and Erland Jonsson. Using active learning in intrusion detection. In *Proceedings. 17th IEEE Computer Security Foundations Workshop, 2004.*, pages 88–98. IEEE, 2004. 9

[18] Matheus Gutoski, Nelson Marcelo Romero Aquino, Manassés Ribeiro, E Lazzaretti, and Heitor Silvério Lopes. Detection of video anomalies using convolutional autoencoders and one-class support vector machines. In *XIII Brazilian Congress on Computational Intelligence*, volume 2017. Rio das Ostras RJ, 2017. 9

[19] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000. 10

[20] Jeroen HM Janssens, Ildikó Flesch, and Eric O Postma. Outlier detection with one-class classifiers from ml and kdd. In *2009 International Conference on Machine Learning and Applications*, pages 147–153. IEEE, 2009. 10

[21] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008. 10

[22] Sahand Hariri, Matias Carrasco Kind, and Robert J Brunner. Extended isolation forest. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1479–1489, 2019. 10

[23] David MJ Tax and Robert PW Duin. Support vector domain description. *Pattern recognition letters*, 20(11-13):1191–1199, 1999. 10

[24] Reza Sadeghi and Javad Hamidzadeh. Automatic support vector data description. *Soft Computing*, 22(1):147–158, 2018. 10

[25] Jorge Rodríguez-Ruiz, Javier Israel Mata-Sánchez, Raul Monroy, Octavio Loyola-González, and Armando López-Cuevas. A one-class classification approach for bot detection on twitter. *Computers & Security*, 91:101715, 2020. 10

[26] Mayank Swarnkar and Neminath Hubballi. Ocpad: One class naive bayes classifier for payload based anomaly detection. *Expert Systems with Applications*, 64:330–339, 2016. 10

[27] Erxue Min, Jun Long, Qiang Liu, Jianjing Cui, Zhiping Cai, and Junbo Ma. Su-ids: A semi-supervised and unsupervised framework for network intrusion detection. In *International Conference on Cloud Computing and Security*, pages 322–334. Springer, 2018. 10

[28] Drausin Wulsin, Justin Blanco, Ram Mani, and Brian Litt. Semi-supervised anomaly detection for eeg waveforms using deep belief nets. In *2010 Ninth international conference on machine learning and applications*, pages 436–441. IEEE, 2010. 10

[29] Jindong Gu, Matthias Schubert, and Volker Tresp. Semi-supervised outlier detection using generative and adversary framework. *arXiv preprint*, 2018. 10

[30] Xianxu Hou, Linlin Shen, Ke Sun, and Guoping Qiu. Deep feature consistent variational autoencoder. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 1133–1141. IEEE, 2017. 10

[31] Ji Feng and Zhi-Hua Zhou. Autoencoder by forest. In *Thirty-Second AAAI conference on artificial intelligence*, 2018. 10

[32] Pramuditha Perera and Vishal M Patel. Learning deep features for one-class classification. *IEEE Transactions on Image Processing*, 28(11):5450–5463, 2019. 10

[33] Mutahir Nadeem, Ochaun Marshall, Sarbjit Singh, Xing Fang, and Xiaohong Yuan. Semi-supervised deep neural network for network intrusion detection. *arXiv preprint*, 2016. 10

[34] Ruoying Wang, Kexin Nie, Tie Wang, Yang Yang, and Bo Long. Deep learning for anomaly detection. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 894–896, 2020. 11

[35] Valentin Leveau and Alexis Joly. Adversarial autoencoders for novelty detection. *arXiv preprint*, 2017. 11

[36] Ashish Mishra, Shiva Krishna Reddy, Anurag Mittal, and Hema A Murthy. A generative model for zero shot learning using conditional variational autoencoders. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 2188–2196, 2018. 11

[37] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010. 11, 15, 49

[38] Qinxue Meng, Daniel Catchpoole, David Skillicom, and Paul J Kennedy. Relational autoencoder for feature extraction. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 364–371. IEEE, 2017. 11

[39] Marcin Pełka. Outlier identification for symbolic data with the application of the dbscan algorithm. In *Conference of the Section on Classification and Data Analysis of the Polish Statistical Association*, pages 53–62. Springer, 2022. 12

[40] Zeyu Wang, Yang Zhang, Qibing Jin, Qie Liu, and Adrian L Kelly. Wiener models robust identification of multi-rate process with time-varying delay using expectation-maximization algorithm. *Journal of Process Control*, 118:126–138, 2022. 12

[41] Meenal Jain, Gagandeep Kaur, and Vikas Saxena. A k-means clustering and svm based hybrid concept drift detection technique for network anomaly detection. *Expert Systems with Applications*, 193:116510, 2022. 12

[42] Imen Souiden, Mohamed Nazih Omri, and Zaki Brahmi. A survey of outlier detection in high dimensional data streams. *Computer Science Review*, 44:100463, 2022. 12

[43] Alexandre LM Levada and Michel FC Haddad. A kullback-leibler divergence-based locally linear embedding method: A novel parametric approach for cluster analysis. In *Brazilian Conference on Intelligent Systems*, pages 406–420. Springer, 2021. 12

[44] Daniella Horan, Eitan Richardson, and Yair Weiss. When is unsupervised disentanglement possible? *Advances in Neural Information Processing Systems*, 34: 5150–5161, 2021. 12

[45] Chi Zhang, Huifen Zhang, Gang Sun, and Xiaoyu Ma. Transformer anomaly detection method based on mds and lof algorithm. In *2022 7th Asia Conference on Power and Electrical Engineering (ACPEE)*, pages 987–991. IEEE, 2022. 12

[46] Simon Bilik and Karel Horak. Feature space reduction as data preprocessing for the anomaly detection. *arXiv preprint arXiv:2203.06747*, 2022. 12

[47] Hongyi Pan, Diaa Badawi, Ishaan Bassi, Sule Ozev, and Ahmet Enis Cetin. Detecting anomaly in chemical sensors via l1-kernels based principal component analysis. *arXiv preprint arXiv:2201.02709*, 2022. 12

[48] Mingqiang Xu, Jun Li, Shuqing Wang, Hong Hao, Huiyuan Tian, and Jie Han. Structural damage detection by integrating robust pca and classical pca for handling environmental variations and imperfect measurement data. *Advances in Structural Engineering*, page 13694332221079090, 2022. 12

[49] Pritam Dey, Zhengwu Zhang, and David B Dunson. Outlier detection for multinetwork data. *arXiv preprint arXiv:2205.06398*, 2022. 12

[50] Liqun Yang, You Zhai, Yipeng Zhang, Yufei Zhao, Zhoujun Li, and Tongge Xu. A new methodology for anomaly detection of attacks in iec 61850-based substation system. *Journal of Information Security and Applications*, 68:103262, 2022. 12

[51] A Ruhan, Xiaodong Mu, and Jingyuan He. Enhance tensor rpca-based mahalanobis distance method for hyperspectral anomaly detection. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022. 13

[52] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Robust, deep and inductive anomaly detection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 36–51. Springer, 2017. 13, 14

[53] Louise Naud and Alexander Lavin. Manifolds for unsupervised visual anomaly detection. *arXiv preprint arXiv:2006.11364*, 2020. 13, 14

[54] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 15, 22, 28, 29, 63, 64

[55] Jiamu Li, Ji Zhang, Jian Wang, Youwen Zhu, Mohamed Jaward Bah, Gaoming Yang, and Yuquan Gan. Vaga: Towards accurate and interpretable outlier detection based on variational auto-encoder and genetic algorithm for high-dimensional data. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 5956–5958. IEEE, 2021. 15

[56] Nitish Srivastava, Ruslan R Salakhutdinov, and Geoffrey E Hinton. Modeling documents with deep boltzmann machines. *arXiv preprint arXiv:1309.6865*, 2013. 15

[57] Andriy Mnih and Danilo Rezende. Variational inference for monte carlo objectives. In *International Conference on Machine Learning*, pages 2188–2196. PMLR, 2016. 15

[58] Sergio Naval Marimont and Giacomo Tarroni. Anomaly detection through latent space restoration using vector quantized variational autoencoders. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 1764–1767. IEEE, 2021. 15, 25, 34

[59] Minyoung Kim, Yuting Wang, Pritish Sahu, and Vladimir Pavlovic. Relevance factor vae: Learning and identifying disentangled factors. *arXiv preprint arXiv:1902.01568*, 2019. 15, 16, 49, 63

[60] Adrian Alan Pol, Victor Berger, Cecile Germain, Gianluca Cerminara, and Maurizio Pierini. Anomaly detection with conditional variational autoencoders. In *2019 18th IEEE international conference on machine learning and applications (ICMLA)*, pages 1651–1657. IEEE, 2019. 15, 25, 30, 63, 67

[61] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006. 15, 26, 49

[62] Douglas A Reynolds. Gaussian mixture models. *Encyclopedia of biometrics*, 741 (659-663), 2009. 15

[63] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 15

[64] Xu Wang, Dezhong Peng, Peng Hu, and Yongsheng Sang. Adversarial correlated autoencoder for unsupervised multi-view representation learning. *Knowledge-Based Systems*, 168:109–120, 2019. 15

[65] Yizhe Zhu, Martin Renqiang Min, Asim Kadav, and Hans Peter Graf. S3vae: Self-supervised sequential vae for representation disentanglement and data generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6538–6547, 2020. 15

[66] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 16

[67] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 16, 49

[68] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016. 16, 32

[69] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017. 16, 23, 49, 63

[70] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 16

[71] Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*, 2018. 16

[72] Li Deng. The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 16, 34, 69

[73] Emile Mathieu, Tom Rainforth, Nana Siddharth, and Yee Whye Teh. Disentangling disentanglement in variational autoencoders. In *International Conference on Machine Learning*, pages 4402–4412. PMLR, 2019. 16, 31, 63, 64

[74] Shuyang Gao, Rob Brekelmans, Greg Ver Steeg, and Aram Galstyan. Auto-encoding total correlation explanation. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1157–1166. PMLR, 2019. 16, 18, 63, 65

[75] Greg Ver Steeg and Aram Galstyan. Discovering structure in high-dimensional data through correlation explanation. *Advances in Neural Information Processing Systems*, 27, 2014. 16

[76] Greg Ver Steeg and Aram Galstyan. Maximally informative hierarchical representations of high-dimensional data. In *Artificial Intelligence and Statistics*, pages 1004–1012. PMLR, 2015. 16

[77] Greg Ver Steeg and Aram Galstyan. Low complexity gaussian latent factor models and a blessing of dimensionality. *arXiv preprint arXiv:1706.03353*, 2017. 16

[78] Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018. 16

[79] Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR, 2020. 16

[80] Hyemi Kim, Seungjae Shin, JoonHo Jang, Kyungwoo Song, Weonyoung Joo, Wanmo Kang, and Il-Chul Moon. Counterfactual fairness with disentangled causal effect variational autoencoder. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8128–8136, 2021. 16

[81] Douglas M Hawkins. *Identification of outliers*, volume 11. Springer, 1980. 22

[82] 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. 22

[83] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 22

[84] Geoffrey E Hinton and Richard Zemel. Autoencoders, minimum description length and helmholtz free energy. *Advances in neural information processing systems*, 6, 1993. 24

[85] Walter Hugo Lopez Pinaya, Sandra Vieira, Rafael Garcia-Dias, and Andrea Mechelli. Autoencoders. In *Machine learning*, pages 193–208. Elsevier, 2020. 24

[86] 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. 24, 27

[87] Bo Du, Wei Xiong, Jia Wu, Lefei Zhang, Liangpei Zhang, and Dacheng Tao. Stacked convolutional denoising auto-encoders for feature representation. *IEEE transactions on cybernetics*, 47(4):1017–1027, 2016. 24

[88] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 665–674, 2017. 24, 33

[89] Xuhong Wang, Ying Du, Shijie Lin, Ping Cui, Yuntian Shen, and Yupu Yang. adVAE: A self-adversarial variational autoencoder with Gaussian anomaly prior knowledge for anomaly detection. *Knowledge-Based Systems*, 190:105187, 2020. 25

[90] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015. 25, 31

[91] Lingheng Meng, Shifei Ding, and Yu Xue. Research on denoising sparse autoencoder. *International Journal of Machine Learning and Cybernetics*, 8(5):1719–1729, 2017. 25, 27

[92] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on Machine Learning*, 2011. 25, 28

[93] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015. 25, 32

[94] Vanessa Böhm and Uroš Seljak. Probabilistic auto-encoder. *arXiv preprint arXiv:2006.05479*, 2020. 25, 32

[95] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 34, 69

[96] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1):1–18, 2015. 35

[97] Andrew L. Beam, Arjun K. Manrai, and Marzyeh Ghassemi. Challenges to the Reproducibility of Machine Learning Models in Health Care. *JAMA*, 323(4):305–306, 01 2020. ISSN 0098-7484. doi: 10.1001/jama.2019.20866. URL `https://doi.org/10.1001/jama.2019.20866`. 36

[98] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. *arXiv preprint arXiv:1611.05148*, 2016. 49

[99] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 63

[100] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. Deep learning for anomaly detection: A review. *ACM Computing Surveys (CSUR)*, 54 (2):1–38, 2021. 63

[101] Matthew D Hoffman, Carlos Riquelme, and Matthew J Johnson. The β-vae's implicit prior. In *Workshop on Bayesian Deep Learning, NIPS*, pages 1–5, 2017. 65

[102] Matthew D Hoffman and Matthew J Johnson. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, volume 1, 2016. 65

[103] Alireza Makhzani and Brendan J Frey. Pixelgan autoencoders. *Advances in Neural Information Processing Systems*, 30, 2017. 65

[104] Milan Studenỳ and Jirina Vejnarová. The multiinformation function as a tool for measuring stochastic dependence. In *Learning in graphical models*, pages 261–297. Springer, 1998. 66, 68

[105] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018. 69

[106] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters (2017). *arXiv preprint arXiv:1702.05373*, 2017. 69

[107] Patrick J Grother. Nist special database 19. *Handprinted forms and characters database, National Institute of Standards and Technology*, 10, 1995. 69

[108] Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. In *International MICCAI brainlesion workshop*, pages 161–169. Springer, 2018. 70

[109] Markus Goldstein and Seiichi Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4):e0152173, 2016. 70

[110] B Ravi Kiran, Dilip Mathew Thomas, and Ranjith Parakkal. An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. *Journal of Imaging*, 4(2):36, 2018. 70

[111] Tim Sainburg, Leland McInnes, and Timothy Q. Gentner. Parametric UMAP Embeddings for Representation and Semisupervised Learning. *Neural Computation*, 33(11):2881–2907, 10 2021. ISSN 0899-7667. doi: 10.1162/neco_a_01434. URL `https://doi.org/10.1162/neco_a_01434`. 71

[112] Antonia Creswell, Yumnah Mohamied, Biswa Sengupta, and Anil A Bharath. Adversarial information factorization. *arXiv preprint arXiv:1711.05175*, 2017. 79

[113] Justin Bayer and Christian Osendorfer. Learning stochastic recurrent networks. *arXiv preprint arXiv:1411.7610*, 2014. 79

[114] Matthew Ashman, Jonathan So, Will Tebbutt, Vincent Fortuin, Michael Pearce, and Richard E Turner. Sparse gaussian process variational autoencoders. *arXiv preprint arXiv:2010.10177*, 2020. 79

[115] Jingyu Yang and Zuogong Yue. Learning hierarchical spatial-temporal graph representations for robust multivariate industrial anomaly detection. *IEEE Transactions on Industrial Informatics*, 2022. 79

[116] Sainan Li, Qilei Yin, Guoliang Li, Qi Li, Zhuotao Liu, and Jinwei Zhu. Unsupervised contextual anomaly detection for database systems. In *Proceedings of the 2022 International Conference on Management of Data*, pages 788–802, 2022. 79