Unsupervised Anomaly Detection for Intricate KPIs via Adversarial Training of VAE

Wenxiao Chen[†§], Haowen Xu[†§], Zeyan Li[†§], Dan Pei[†§*], Jie Chen[‡], Honglin Qiao[‡], Yang Feng[‡], Zhaogang Wang[‡]
[†]*Department of Computer Science and Technology, Tsinghua University* [‡] *Alibaba Group*
[§] *Beijing National Research Center for Information Science and Technology (BNRist)*

***Abstract*—To ensure the reliability of the Internet-based application services, KPIs (Key Performance Monitors) are closely monitored in real time and the anomalies presented in the KPIs must be discovered in time. While anomaly detection for the seasonal smooth service-level KPIs (*e.g.*, number of transactions per minute) have been solved reasonably well in the literature, the intricate KPIs at the machine level (*e.g.*, the number of I/O requests on a server monitored per second) has been little studied. These intricate KPIs are prevalent and important, but exhibit non-Gaussian noises and complex data distribution that are hard to model. In this paper, we propose an adversarial training method in the Bayesian network based on partition analysis with solid theoretical proof. Based on it, we propose the first unsupervised anomaly detection algorithm *Buzz* for intricate KPIs with high performance. Its best F-scores on the data from a global Internet company range from 0.92 to 0.99, significantly outperforming a state-of-art VAE-based unsupervised approach without adversarial training and a state-of-art supervised approach.**

I. Introduction

To ensure the reliability of the Internet-based application services, KPIs (Key Performance Indicators) are closely monitored in real time. When KPIs show anomalies (such as sudden increase, sudden drop, and jitter), some potential failures have occurred in the related applications [1], [2]. In order to reduce the cost of failures, anomalies presented in KPIs must be discovered accurately in time.

While anomaly detection for the seasonal smooth service-level KPIs (*e.g.*, number of transactions per minute) have been solved reasonably well [3]–[5], the intricate KPIs at the machine level (*e.g.*, the number of I/O requests on a server monitored per second) have been little studied. These intricate KPIs are prevalent and important, but exhibit non-Gaussian noises and complex data distribution that are hard to model. Fig. 1 shows a few such KPIs on which Opprentice [3] (a state-of-art supervised approach) and Donut [4] (a state-of-art unsupervised approach based on variational auto-encoder (VAE)) do not perform well, as will be shown in § V.

In this paper, we propose an adversarial training method for VAE based on partition analysis. Based on it, we propose the first unsupervised anomaly detection algorithm *Buzz* for intricate KPIs with high performance.

Buzz has a few key ideas. First, to make the modeling of intricate KPIs tractable, we apply partitioning, a common analysis method in measure theory. More specifically, we divide the data space into several subspaces (partitions) and calculate the distance in each subspace. Second, when calculating the distance, we use Wasserstein distance [6] between generative distribution and empirical distribution, which has been shown in WGAN [6] to be a robust metric in distribution space.

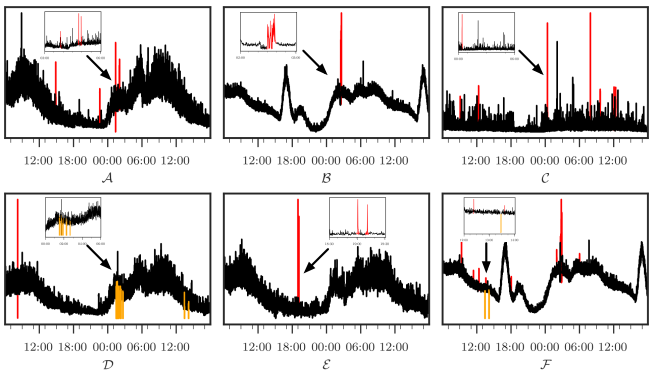


Fig. 1: Intricate KPIs in this paper, each of which is plotted with a 36-hour-long segment, with anomalies in red and missing in yellow. The small pictures show the detailed KPIs nearby the missing and anomalies. The arrows indicate position of small pictures in the large pictures.

Third, we propose a *primal* form of training objective with theoretical deduction, and then transform our model into a Bayesian network. In particular, *Buzz* essentially optimizes the evidence lower bound of likelihood of a variant of VAE by adversarial training. Fourth, we use VAE as generative model to generate samples and use another neural network as discriminative model to distinguish generative samples and real samples. Fifth, to ensure that the adversarial training is stable, we adopt the gradient penalty technique [7], an improvement over the original training method from WGAN [6]. Finally, anomaly detection is conducted by Bayesian inference.

The contributions of *Buzz* are summarized as follows.

- *Buzz* is the first unsupervised anomaly detection algorithm via deep generative model on intricate KPIs. *Buzz*'s best F-scores on the data from a top global Internet company range from 0.92 to 0.99, significantly outperforming existing approaches.
- The training method proposed in *Buzz* is the first adversarial training method for VAE, based on partitions analysis with solid theoretical deduction and experimental

* Dan Pei is the corresponding author.

support.

- We propose a *primal* form of training objective of *Buzz* from Wasserstein distance based on partition analysis and give theoretical deduction to transform our model into a Bayesian network. It is a novel idea to build the bridge between Bayesian networks and optimal transport theory.

II. Background and Problem

A. KPI Anomaly Detection

A **KPI** is a time series, and can be denoted as $X = \{x_1, x_2, \cdots, x_T\}$, where x_t is the value corresponding to time index t for $t \in \{1, 2, \cdots, T\}$.

Anomaly Detection on a KPI is to determine whether the value x_t is an anomaly given a recent history of W data points. If so, $\alpha_t = 1$. An anomaly detection algorithm typically computes the conditional probability, $P(\alpha_t = 1 | x_{t-W+1}, \ldots, x_t)$, instead of directly giving the value of α_t. Therefore, fundamentally any KPI anomaly detection algorithm needs to somehow model this conditional probability distribution.

B. Intricate KPIs

In this paper, we focus on the anomaly detection on intricate KPIs. KPIs can be roughly divided into 2 types: the seasonal smooth KPIs and intricate KPI. The former is usually statistics at the service/business levels (*e.g.*, number of transactions per minute). One can roughly assume that these KPIs have diagonal multivariate Gaussian noises. Intricate KPIs are usually lower-level (*e.g.*, number of I/O requests per second on server of a distributed database). Intricate KPIs are often monitored in a fine granularity in order to catch the micro-congestion caused by the bursty traffic (*e.g.*, typical of database traffic). One can roughly assume that the noises in intricate KPIs are not diagonal multivariate Gaussian.

Fig. 1 shows a few examples of intricate KPIs. As can be seen, intricate KPIs are complex, jitter violently at short time scale, yet globally there appear to be some patterns. Furthermore, different intricate KPIs can have different global and local patterns. Thus, it is challenging to precisely define intricate KPIs or enumerate different types of intricate KPIs. As such, it is intractable to design a framework and test it for all intricate KPIs. Therefore, in this paper we focus on the intricate KPIs that we countered, and are important in practice. More specifically, we obtain 11 well-maintained intricate KPIs from a large Internet company with manual anomaly labels, and we show part of them in Fig. 1. They represent a series of important, practical and intricate KPIs. The operators that we worked with confirm that it is of urgent practical significance to solve the anomaly detection on these intricate KPIs.

C. Previous Anomaly Detection Approaches

Many anomaly detectors based on traditional statistical models have been proposed over the years, *e.g.*, [8] *et al.* [9]–[14], but algorithm selection and parameter tuning needs to be done on a per-KPI basis, and they cannot capture the complex data distribution in intricate KPIs.

More recent methods use supervised ensemble learning with above detectors as features, such as EGADS [15] and Opprentice [3], and showed promising results on smooth KPIs. However, their labeling overhead is too large, and their features (from traditional statistical models) are not appropriate for intricate KPIs.

Unsupervised anomaly detection approaches *e.g.*, [16]–[20] learn to earn the normal data pattern and derive the conditional probability $P(\alpha_t = 1 | x_{t-W+1}, \ldots, x_t)$ from the normal data pattern by assumption for anomalies, *e.g.*, the likelihood of anomalies is negligible. *Donut* [4] is the state-of-art unsupervised anomaly detection approach. It is based on VAE [21], [22], with high performance and solid theoretical analysis on seasonal smooth KPI. But because *Donut* assumes diagonal multivariate Gaussian noises, it does not perform well on intricate KPIs, as will be shown in Fig. 9.

D. Variational Auto-Encoder

VAE [21], [22] is a deep Bayesian network which models the relationship between two random variables $\mathbf{x}$ and $\mathbf{z}$. $p(\mathbf{x})$ is called empirical distribution and $p(\mathbf{z})$ is called prior distribution, usually multivariate standard Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The form of conditional distribution $p_\theta(\mathbf{x}|\mathbf{z})$ is chosen according to the particular requirement of task. Then, $p_\theta(\mathbf{x}) = \mathbb{E}_{p(\mathbf{z})}\left[p_\theta(\mathbf{x}|\mathbf{z})\right]$ can be seen as a kind of kernel density estimation. $q_\phi(\mathbf{z}|\mathbf{x})$ is an approximation posterior of the true posterior $p_\theta(\mathbf{z}|\mathbf{x})$ which is intractable. $q_\phi(\mathbf{z}|\mathbf{x})$ can be fitted by neural network through maximizing the evidence lower bound of likelihood (ELBO) with SGVB algorithm.

The training objective of VAE, denoted by $\mathcal{L}_{vae}$, is the evidence lower bound of $\mathbb{E}_{p(\mathbf{x})}\left[\log p_\theta(\mathbf{x})\right]$.

$$\mathcal{L}_{vae} = \mathbb{E}_{p(\mathbf{x})}\left[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathrm{KL}\left[q_\phi(\mathbf{z}|\mathbf{x}) \,\middle\|\, p_\theta(\mathbf{z})\right]\right]$$

Donut [4] modified a part of ELBO to avoid the influence of anomaly in training and achieved high performance on seasonal smooth KPIs. However, *Donut* doesn't work well on intricate KPIs — we train *Donut* for several times and find that the performance of *Donut* is low, unstable, and it is not well-trained (shown later in § V). We conjecture that it is hard to train on intricate empirical distribution $p(\mathbf{x})$ because of the limit of neural network expressing capacity and training method with limited training samples with support of Fig. 8.

E. Adversarial Training

A series of adversarial training methods have been proposed, such as GAN [23], WGAN [6], AAE [24], WAE [25], and GAN-OT [26]. In adversarial training, a generator model tries to generate samples to deceive a discriminator model, and the discriminator tries to distinguish the generated samples and real samples. During the adversarial training, the ability of both generator and discriminator can greatly improve. It has been shown that adversarial training achieves great performance on complex empirical distribution in image classification, image generation, speech recognition and other domains.

There are several studies about combining VAE and adversarial training, such as [27], similar to our structure at the

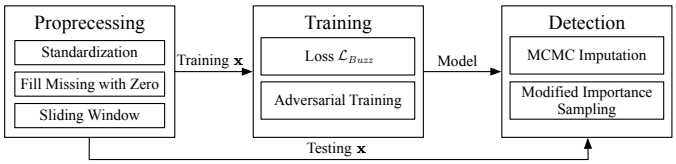


Fig. 2: Overall architecture of *Buzz*.

first glance, but our theoretical proof shows they are totally different in essence. AAE [24] proposes an adversarial training on the prior distribution $p(\mathbf{z})$ with high performance and solid proof. Inspired by it, we try to propose an adversarial training method on intricate empirical distribution $p(\mathbf{x})$ for VAE. Based on it, an anomaly detection algorithm for intricate KPIs via deep generative model is proposed in this paper.

III. Architecture

In this section, we will introduce our motivation and proposed framework, *Buzz*, for anomaly detection, including the preprocessing stage, the training objective and corresponding algorithm, the neural network architecture, as well as the detection method. The overall architecture is shown in Fig. 2.

A. Motivation

There are two major ideas in *Buzz*: Wasserstein distance and Partitioning from measure theory.

When calculating distance, we use Wasserstein distance [6] between generative distribution and empirical distribution (called *distribution distance* hereinafter), which has been shown in WGAN [6] to be robust when measuring the distance between probability distribution.

Partitioning is a powerful and commonly used analysis method for distribution in measure theory [28], [29]. The basic idea is in spirit similar to a common technique in calculus: when we calculate the integral of a complicated function, we often divide its integral domain into several partitions and calculate the integral on each partition, then get the average of them. Similarly, we divide the space $\mathcal{X}$ with intricate empirical distribution into several partitions, and intuitively it may become easier to calculate distribution distance on each small enough partition than on the whole space.

The distribution distance on each partition is calculated by adversarial training, and the global distance is the expectation of distribution distance on all the partitions, as shown in Fig. 3.

Coincidentally, we notice that when each partition is smaller and smaller, the global distance approaches the reconstruction term in the evidence lower bound of a special variant of VAE, whose posterior distribution is an exponential distribution. Partition plays the role connecting the loss of WGAN and VAE during the partitions changing from the whole to the point-wise. It inspires our adversarial training method for VAE.

We will give the theoretical deduction of this motivation, and an approximation training objective in § IV. In this section, we will first demonstrate how it works in practice.

B. Preprocessing

The KPIs in real applications are complex time-series data. Sometimes a value is not captured by monitors and is set to

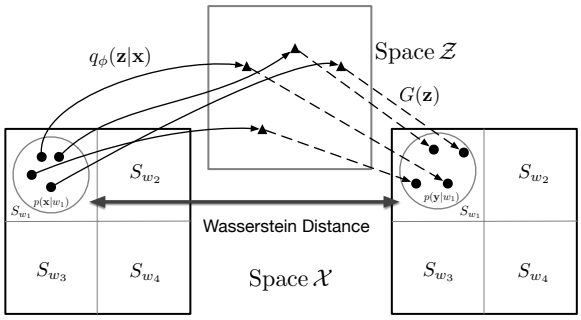
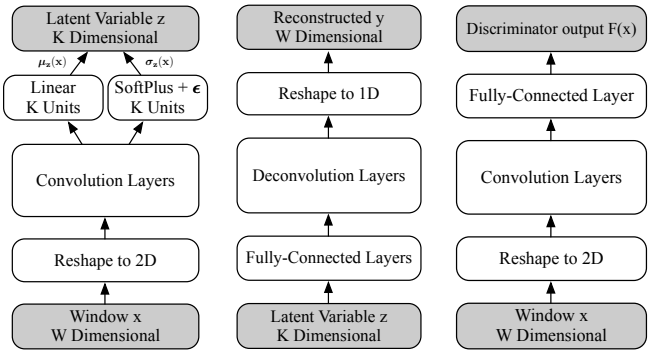


Fig. 3: An example for our motivation, where $\mathcal{X}$ is divided into 4 partitions. For partition S_{w_1}, $p(\mathbf{x}|w_1)$ is obtained by restricting $p(\mathbf{x})$ on S_{w_1} and normalizing it. $p_G(\mathbf{y}|w_1)$ is the generative distribution from $p(\mathbf{x}|w_1)$ by $q_\phi(\mathbf{z}|\mathbf{x})$ and $G(\mathbf{z})$. We calculate the Wasserstein distance between $p(\mathbf{x}|w_1)$ and $p_G(\mathbf{y}|w_1)$ as the distribution distance on S_{w_1}, instead of calculating the distance between pair $(\mathbf{x}, \mathbf{y})$ in ELBO. Then we compute the distribution distance on S_{w_2}, S_{w_3} and S_{w_4} by the same method. The global distance is the expectation of these distribution distance.



(a) Variational net (b) Generative net (c) Discriminative net

Fig. 4: Network structure of *Buzz*. Gray nodes are random variables, and white nodes are layers.

NaN, which is called missing. Sometimes the scale of the values are all very large for a period of time. These values will bring trouble to the training and detection, thus we need to preprocess the data.

Firstly, we set the missing values to zeros and split the KPI into training set and testing set. Secondly, we measure the mean μ and variance σ over the training set. Thirdly, we standardize the data by setting each value x to be $\frac{(x-\mu)}{\sigma}$. Fourthly, we truncate the standardized values to $[-10, 10]$.

The inputs of our model are sliding windows taken from the standardized KPI, and each window is a W-long time series segment where W is a hyper-parameter called window size. The window ending at time t is denoted by $\mathbf{x}^{(t)}$ and the k-th value in the window $\{x_{t-W+1}, \ldots, x_t\}$ is denoted by $\mathbf{x}_k^{(t)}$.

C. Neural Network

Our model consists of 3 sub-networks, the *variational network*, the *generative network* and the *discriminative network*, as shown in Fig. 4a, Fig. 4b and Fig. 4c.

The *variational network* is designed to find the corresponding pattern $q_\phi(\mathbf{z}|\mathbf{x})$ from a given window $\mathbf{x}$. We reshape the

window into a 2D matrix and use convolution layers [30] to extract its high-level features, which are denoted by $h_{\mathbf{z}}(\mathbf{x})$. Then we derive the mean and standard deviation of $q_\phi(\mathbf{z}|\mathbf{x})$ by: $\boldsymbol{\mu}_{\mathbf{z}}(\mathbf{x}) = W_{\boldsymbol{\mu}_{\mathbf{z}}}^\top \cdot h_{\mathbf{z}}(\mathbf{x}) + b_{\boldsymbol{\mu}_z}$ and $\boldsymbol{\sigma}_{\mathbf{z}}(\mathbf{x}) = \mathrm{SoftPlus}(W_{\boldsymbol{\sigma}_{\mathbf{z}}}^\top \cdot h_{\mathbf{z}}(\mathbf{x}) + b_{\boldsymbol{\sigma}_{\mathbf{z}}}) + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is a small positive constant vector.

The *generative network* is designed to generate a reconstructed window for $q_\phi(\mathbf{z}|\mathbf{x})$, given by the *variational network*. We map $\mathbf{z}$ to a 2D matrix by fully-connected layers, then pass it through a series of 2D transposed convolutional layers, and finally reshape it to 1D, to get the reconstructed window $G(\mathbf{z})$.

The *discriminative network* is designed to distinguish real window $\mathbf{x}$ from reconstructed window $\mathbf{y}$. We reshape the window into 2D, obtain high-level features by convolution layers, pass the features through fully-connected layer, and finally obtain the discriminator output $F(\mathbf{x})$.

D. Training Objective

The most important part of *Buzz* is its training objective. We propose a novel training objective $\tilde{\mathcal{L}}_{Buzz}$ for intricate KPIs to solve the problem that it is hard to train models on our dataset using pure Bayesian lower-bounds, like the *Donut* approach (see § V). The precise definition and deduction of $\tilde{\mathcal{L}}_{Buzz}$ are given in § IV. In this section, we only give the sampling form of $\tilde{\mathcal{L}}_{Buzz}$ and the training algorithm for it.

Symbols s, b are parameters, the neighborhood size and the batch size. Let $\mathcal{W}$ be $\{w_1, w_2, \ldots, w_b\}$, a mini-batch of randomly selected time, satisfying that each w_i is a multiple of s, and $w_i \neq w_j \ \forall i \neq j$. We call this condition on $\mathcal{W}$, the neighborhood condition (NC). The neighborhood set for $w \in \mathcal{W}$ is $\{w, w+1, \ldots, w+s-1\}$, which is a partition on time. The union of Voronoi cells of $\mathbf{x}^{(w)}, \mathbf{x}^{(w+1)}, \ldots, \mathbf{x}^{(w+s-1)}$, is a partition S_w on space $\mathcal{X}$. It is a simple efficient partition method. Define symbols:

$$\mathcal{K} = \frac{1}{bs} \sum_{w \in \mathcal{W}} \sum_{i=0}^{s-1} \mathrm{KL}\left[q_\phi(\mathbf{z}|\mathbf{x}^{(w+i)}) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{1}) \right]$$

$$Z(\lambda) = \frac{\Gamma(W)}{\Gamma(\frac{W}{2})} 2\pi^{\frac{W}{2}} \lambda^{-W}, \quad \Gamma \text{ is the Gamma function}$$

$$\mathcal{T}(F, w) = \frac{1}{bs} \sum_{i=0}^{s-1} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}^{(w+i)})} \left[F(\mathbf{x}^{(w+i)}) - F(G(\mathbf{z})) \right]$$

$$\mathcal{R}(F, w) = \frac{1}{bs} \sum_{i=0}^{s-1} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}^{(w+i)})} \left[\mathbb{E}_{\xi \sim [0,1]} (\|\nabla_{\hat{\mathbf{x}}} F(\hat{\mathbf{x}})\| - \mathbf{1})^2 \right]$$

where $\hat{\mathbf{x}}$ denotes $\xi \mathbf{x}^{(w+i)} + (1-\xi) G(\mathbf{z})$. Then the training objective $\tilde{\mathcal{L}}_{Buzz}$ can be given by

$$\tilde{\mathcal{L}}_{Buzz} = -\lambda \sup_F [\sum_{w \in \mathcal{W}} (|\mathcal{T}(F, w)| - \eta \mathcal{R}(F, w))] - \mathcal{K} - \log Z(\lambda)$$

$\tilde{\mathcal{L}}_{Buzz}$ is an improvement of the loss in WGAN-GP [7], a special adversarial training algorithm. The *discriminative network* ($F(\mathbf{x})$) in our model can be seen as the "discriminator" of WGAN-GP, while the *variational network* and the *generative network* can be seen as the "generator". $\sup_F[\cdot]$ and $\mathcal{T}(F, w)$ can be seen as the major "WGAN" loss term. $\mathcal{R}(F, w)$ can be seen as the regularizer for F, *i.e.*, the "-GP" (gradient penalty) term, while η is the gradient penalty weight.

In addition to the terms used in WGAN-GP [7], $\tilde{\mathcal{L}}_{Buzz}$ also adds the term $\mathcal{K}$, borrowed from Bayesian training objectives, serving as the regularizer for $q_\phi(\mathbf{z}|\mathbf{x})$. λ is a trainable variable, induced from the Bayesian inference framework, which balances the WGAN-GP terms and the Bayesian regularizer.

E. Training

Given $\tilde{\mathcal{L}}_{Buzz}$ is an improvement loss of WGAN-GP, so *Buzz*'s training procedure also shares some similarities with the WGAN-GP algorithm. The parameters of the "generator" (*i.e.*, the parameters of the *variational network* and the *generative network*, plus λ) are denoted by ω, while the parameters of the "discriminator" (*i.e.*, those of the *discriminative network* $F(\mathbf{x})$), are denoted by ν. The $\mathcal{R}(F, w)$ term is ignored when optimizing ω, since it is just a regularizer for F, depending only on ν. We use SGVB [21] to solve the variational inference for $q_\phi(\mathbf{z}|\mathbf{x})$, and Adam [31] to optimize the network parameters. Thanks to the strong convergence property of the WGAN-GP loss, we enjoy a very stable training process with few hyper-parameters tuning.

It will be proven in § IV that $\mathcal{L}_{Buzz} \to \mathcal{L}_{vae}$ when $s \to 1$($\mathcal{L}_{Buzz}$ is the *primal* form that $\tilde{\mathcal{L}}_{Buzz}$ approximates). Therefore, we can turn our model into a Bayesian network after training, which is required by § III-F. Thus, in algorithm 1, we set $s = s_0$ at beginning, and gradually decrease s down to 1, by setting $s \leftarrow s/2$ after every few epochs.

F. Detection

We shall build a bridge between $\mathcal{L}_{Buzz}$ and $\mathcal{L}_{vae}$, the loss of a special variant of VAE [21], in § IV, by letting $p_\theta(\mathbf{x}|\mathbf{z}) = \frac{1}{Z(\lambda)} \exp\{-\lambda \|\mathbf{x} - G(\mathbf{z})\|\}$. With this technique, we can turn our model, trained by algorithm 1 (a variant of the WGAN-GP algorithm), into a Bayesian network. We can then derive the detection output by the probabilistic framework, as follows.

When a new point is to be detected, the last window (*i.e.*, whose last data point is the new point) is denoted by $\mathbf{x}$. Since our goal is to detect whether the last data point is an anomaly, we assume it to be "anomaly", and iteratively use MCMC imputation (also used by *Donut* [4]) to obtain a reasonable estimation of the reconstructed $\bar{\mathbf{x}}$, by following procedure:

1) $\bar{\mathbf{x}} \leftarrow \mathbf{x}$, and $\boldsymbol{\alpha}_i = \begin{cases} 1, & i = W \text{ or } \mathbf{x}_i \text{ is missing} \\ 0, & \text{otherwise} \end{cases}$

2) Repeat for T_{MC} times: $\bar{\mathbf{x}} \leftarrow (1 - \boldsymbol{\alpha}) \odot \bar{\mathbf{x}} + \boldsymbol{\alpha} \odot G(\mathbf{z})$

Finally, we use $\log p_\theta(\mathbf{x}) - \log p_\theta(\bar{\mathbf{x}})$ as the anomaly score for the last point, which is computed by:

$$\log \frac{1}{L} \sum_{l=1}^{L} \left[\frac{p_\theta(\mathbf{x}|\mathbf{z}^{(l)}) \, p_\theta(\mathbf{z}^{(l)})}{q_\phi(\mathbf{z}^{(l)}|\bar{\mathbf{x}})} \right] - \log \frac{1}{L} \sum_{l=1}^{L} \left[\frac{p_\theta(\bar{\mathbf{x}}|\mathbf{z}^{(l)}) \, p_\theta(\mathbf{z}^{(l)})}{q_\phi(\mathbf{z}^{(l)}|\bar{\mathbf{x}})} \right]$$

where $\mathbf{z}^{(l)} \sim q_\phi(\mathbf{z}|\bar{\mathbf{x}})$, and L is the sampling number. The formula is slightly modified from the importance sampling [32] based estimation of $\log p_\theta(\mathbf{x}) - \log p_\theta(\bar{\mathbf{x}})$ (denoted by the *Buzz-strict* detector). We use $q_\phi(\mathbf{z}|\bar{\mathbf{x}})$ to replace $q_\phi(\mathbf{z}|\mathbf{x})$ when computing the importance sampling formula for $\log p_\theta(\mathbf{x})$, because $q_\phi(\mathbf{z}|\mathbf{x})$ may deviate due to the influence of anomaly. Experiment results in Fig. 6 confirm this conjecture. The

Algorithm 1: *Buzz* training

Require: The gradient penalty weight η, the number of critic iterations n_{critic}, the initial neighborhood size s_0 and batch size b_0. The parameters for Adam Optimizer, α_0, β_1, β_2.

Initial the parameters ω, ν, $s = s_0, b = b_0$.
repeat
  repeat
    for $t = 1, \ldots, n_{critic} + 1$ **do**
      Sample $w_1 \ldots w_b$ s.t. NC.
      $\mathcal{L}_\nu \leftarrow 0, \mathcal{L}_\omega \leftarrow 0$
      for $i = 1 \ldots b$ **do**
        $\mathcal{L}_i \leftarrow 0, \mathcal{L}_i^{(\eta)} \leftarrow 0, \mathcal{L}_i^{(K)} \leftarrow 0$
        for $j = 0 \ldots s-1$ **do**
          Set $\mathbf{x} \leftarrow \mathbf{x}^{(w_i+j)}$
          Obtain $\boldsymbol{\iota} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$, set $\mathbf{z} \leftarrow \boldsymbol{\iota} \odot \boldsymbol{\sigma}_{\mathbf{z}}(\mathbf{x}) + \boldsymbol{\mu}_{\mathbf{z}}(\mathbf{x})$.
          Obtain $\xi \sim [0, 1]$, set $\hat{\mathbf{x}} \leftarrow \xi\mathbf{x} + (1-\xi)G(\mathbf{z})$.
          $\mathcal{L}_i \leftarrow \mathcal{L}_i + F(\mathbf{x}) - F(G(\mathbf{z}))$
          $\mathcal{L}_i^{(\eta)} \leftarrow \mathcal{L}_i^{(\eta)} + \eta(\|\nabla_{\hat{\mathbf{x}}} F(\hat{\mathbf{x}})\| - 1)^2$
          $\mathcal{L}_i^{(K)} \leftarrow \mathcal{L}_i^{(K)} + \mathrm{KL}\left[\mathcal{N}(\mu_{\mathbf{z}}(\mathbf{x}), \sigma_{\mathbf{z}}(\mathbf{x})) \,\middle\|\, \mathcal{N}(\mathbf{0}, \mathbf{1})\right]$
        end
        $\mathcal{L}_\nu \leftarrow \mathcal{L}_\nu + |\mathcal{L}_i| - \mathcal{L}_i^{(\eta)}$
        $\mathcal{L}_\omega \leftarrow \mathcal{L}_\omega - \lambda|\mathcal{L}_i| - \mathcal{L}_i^{(K)} - \log Z(\lambda)$
      end
      if $t = n_{critic} + 1$ **then**
        $\omega \leftarrow \mathrm{Adam}(\nabla_\omega \frac{-1}{bs}\mathcal{L}_\omega, \omega, \alpha_0, \beta_1, \beta_2)$
      else
        $\nu \leftarrow \mathrm{Adam}(\nabla_\nu \frac{-1}{bs}\mathcal{L}_\nu, \nu, \alpha_0, \beta_1, \beta_2)$
      end
    end
  until ω convergence;
  $s \leftarrow \frac{s}{2}, b \leftarrow 2b$
until $s = 0$;

threshold for detecting anomaly from the anomaly score in practice is selected by the best F-score as [4] do.

IV. Theorem

In this section, we will give the theoretical deduction of $\mathcal{L}_{Buzz}$, the training objective of *Buzz*, and we shall explain why we can turn our model, trained by algorithm 1, into a Bayesian network. The content of this section is divided into four parts: (1) we define symbols, useful in deduction; (2) we build a bridge from the *primal* form of $\mathcal{L}_{Buzz}$, to $\mathcal{L}_{vae}$, the loss of a special variant of VAE; (3) we explain how we can rewrite the *primal* form of $\mathcal{L}_{Buzz}$, into its *dual* form; and finally (4) we give an approximation $\tilde{\mathcal{L}}_{Buzz}$, to the *dual* form $\mathcal{L}_{Buzz}$, which is relatively easy to compute, and is used in algorithm 1.

A. Notation

The VAE training objective, using SGVB [33], is given as:

$$\mathcal{L}_{vae} = \mathbb{E}_{p(\mathbf{x})}\left[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathrm{KL}\left[q_\phi(\mathbf{z}|\mathbf{x}) \,\middle\|\, p_\theta(\mathbf{z})\right]\right]$$

We present a special variant of VAE here, to help induce the theorems. The prior distribution $p_\theta(\mathbf{z})$ is chosen to be a K-dimensional unit Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$, while the posterior $p_\theta(\mathbf{x}|\mathbf{z})$ is chosen to be $\frac{1}{Z(\lambda)}\exp\{-\lambda\|\mathbf{x} - G(\mathbf{z})\|\}$. G is a deterministic function. λ is a constant. $Z(\lambda)$ depends on λ and W, the dimension of $\mathbf{x}$. $\mathfrak{S}_{W-1} = 2\pi^{\frac{W}{2}}/\Gamma(\frac{W}{2})$ is the surface area of the unit $(W-1)$-sphere. It is not hard to show:

$$Z(\lambda) = \int_0^\infty \mathfrak{S}_{W-1} r^{W-1} e^{-\lambda r} \mathrm{d}r = \mathfrak{S}_{W-1}\Gamma(W)\lambda^{-W}$$

We denote the Euclid space of $\mathbf{x}$ by $\mathcal{X}$, while the latent space of $\mathbf{z}$ by $\mathcal{Z}$. Define the partitions of $\mathcal{X}$ to be $\{S_w \big| S_w \text{ is Lebesgue measurable set}\}$, *s.t.* $\sqcup_w S_w = \mathcal{X}$, where $\sqcup$ means disjoint union. Define $S = \{(\mathbf{x}_1, \mathbf{x}_2) \big| \exists w, \mathbf{x}_1 \in S_w, \mathbf{x}_2 \in S_w\}$.

Define $p(\mathbf{x}, w) = p(\mathbf{x})1_{S_w}(\mathbf{x})$, where $1_{S_w}(\mathbf{x}) = 1$ if $\mathbf{x} \in S_w$, otherwise 0. Then $p(w) = \int_{\mathcal{X}} p(\mathbf{x}, w)\mathrm{d}\mathbf{x} = \int_{S_w} p(\mathbf{x})\mathrm{d}\mathbf{x}$, and $p(\mathbf{x}|w) = p(\mathbf{x}, w)/p(w)$. Both $p(w)$ and $p(\mathbf{x}|w)$ are well-defined according to the property of Lebesgue measurable sets.

Define the conditional distribution $p_G(\mathbf{y}|\mathbf{z}) = \delta(\mathbf{y} - G(\mathbf{z}))$, a dirac distribution, then $p_G(\mathbf{y}|\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[p_G(\mathbf{y}|\mathbf{z})]$, and $p_G(\mathbf{y}|w) = \mathbb{E}_{p(\mathbf{x}|w)}[p_G(\mathbf{y}|\mathbf{x})]$. Notice that $\mathbf{x}$ and $\mathbf{z}$ are "true" variables of our final Bayesian net, while $\mathbf{y}$ and w are just auxiliary variables, helping to induce the theorems.

We use $A \downarrow$ to denote a decreasing sequence $\{A^{(n)}\}$, *s.t.*, $A^{(n+1)} \leq A^{(n)}$ if $A^{(n)} \in \mathbb{R}$, or $A^{(n+1)} \subseteq A^{(n)}$ if $A^{(n)}$ are sets. We use $A \downarrow B$ to denote $A \downarrow$ and $\lim_{n\to\infty} A^{(n)} = B$.

The *primal* form of the *Buzz* training objective is given by:

$$\mathcal{L}_{Buzz} = -\lambda\,\mathbb{E}_{p(w)}\, W^1[P(\mathbf{x}|w)\|P_G(\mathbf{y}|w)] - \mathcal{K} - \log Z(\lambda)$$

where $W^1[P(\mathbf{x}|w)\|P_G(\mathbf{y}|w)]$ is 1-th Wasserstein distance between the two distributions $P(\mathbf{x}|w)$ and $P_G(\mathbf{y}|w)$, and $\mathcal{K} = \mathbb{E}_{p(\mathbf{x})}\left[\mathrm{KL}\left[q_\phi(\mathbf{z}|\mathbf{x}) \,\middle\|\, p_\theta(\mathbf{z})\right]\right]$.

Define $\Gamma(P(\mathbf{x}|w), P_G(\mathbf{y}|w)) = \{\gamma(\mathbf{x}, \mathbf{y}) \big| \int_{\mathcal{X}} \gamma(\mathbf{x}, \mathbf{y})\mathrm{d}\mathbf{y} = p(\mathbf{x}|w), \int_{\mathcal{X}} \gamma(\mathbf{x}, \mathbf{y})\mathrm{d}\mathbf{x} = p_G(\mathbf{y}|w)\}$, denoted by Γ_w. We call $\gamma \in \Gamma_w$, a *coupling* of $P(\mathbf{x}|w), P_G(\mathbf{y}|w)$. The *primal* form of $W^1[P(\mathbf{x}|w)\|P_G(\mathbf{y}|w)]$ can then be given by:

$$W^1[P(\mathbf{x}|w)\|P_G(\mathbf{y}|w)] = \inf_{\gamma\in\Gamma_w} \int_{\mathcal{X}\times\mathcal{X}} \|\mathbf{x} - \mathbf{y}\|\mathrm{d}\gamma(\mathbf{x}, \mathbf{y})$$

A special case of the duality theorem of Kantorovich and Rubinstein [34] gives the *dual* form of $W^1[P(\mathbf{x}|w)\|P_G(\mathbf{y}|w)]$:

$$W^1[P(\mathbf{x}|w)\|P_G(\mathbf{y}|w)] = \sup_{Lip(f)\leq 1} \left\{\int_{\mathcal{X}} f(\mathbf{x})(p(\mathbf{x}|w)\mathrm{d}\mathbf{x} - \int_{\mathcal{X}} f(\mathbf{y})(p_G(\mathbf{y}|w)\mathrm{d}\mathbf{y}\right\}$$

B. From $\mathcal{L}_{Buzz}$ to $\mathcal{L}_{vae}$

Lemma IV.1.

$$\mathcal{L}_{vae} = \lambda\,\mathbb{E}_{p(w)}\left[\mathbb{E}_{p(\mathbf{x}|w)}\,\mathbb{E}_{p_G(\mathbf{y}|\mathbf{x})} - \|\mathbf{x} - \mathbf{y}\|\right] - \mathcal{K} - \log Z(\lambda)$$

Proof. Given $p(\mathbf{x}|\mathbf{z}) = \frac{1}{Z(\lambda)}\exp\{-\lambda\|\mathbf{x} - G(\mathbf{z})\|\}$, we have:

$$\begin{aligned}
\mathcal{L}_{vae} &= \mathbb{E}_{p(\mathbf{x})}\left[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \mathcal{K} \\
&= \mathbb{E}_{p(w)}\left[\mathbb{E}_{p(\mathbf{x}|w)}\,\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \mathcal{K} \\
&= \lambda\,\mathbb{E}_{p(w)}\left[\mathbb{E}_{p(\mathbf{x}|w)}\,\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} - \|\mathbf{x} - G(\mathbf{z})\|\right] - \mathcal{K} - \log Z(\lambda)
\end{aligned}$$

The following equation is provided in [25] and [35]. We demonstrate it here, for the completeness of our proof. Notice that $\mathbb{E}_{p_G(\mathbf{y}|\mathbf{z})} \|\mathbf{x} - G(\mathbf{z})\| = \mathbb{E}_{p_G(\mathbf{y}|\mathbf{z})} \|\mathbf{x} - \mathbf{y}\|$, since $p_G(\mathbf{y}|\mathbf{z})$ is dirac; and that we exchange the order of the integrations by Fubini theorem [36]:

$$\begin{aligned}
&\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \|\mathbf{x} - G(\mathbf{z})\| = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \mathbb{E}_{p_G(\mathbf{y}|\mathbf{z})} \|\mathbf{x} - G(\mathbf{z})\| \\
&= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \mathbb{E}_{p_G(\mathbf{y}|\mathbf{z})} \|\mathbf{x} - \mathbf{y}\| = \iint q_\phi(\mathbf{z}|\mathbf{x}) p_G(\mathbf{y}|\mathbf{z}) \|\mathbf{x} - \mathbf{y}\| \mathrm{d}\mathbf{y} \mathrm{d}\mathbf{z} \\
&= \int \left(\int q_\phi(\mathbf{z}|\mathbf{x}) p_G(\mathbf{y}|\mathbf{z}) \mathrm{d}\mathbf{z} \right) \|\mathbf{x} - \mathbf{y}\| \mathrm{d}\mathbf{y} = \mathbb{E}_{p_G(\mathbf{y}|\mathbf{x})} \|\mathbf{x} - \mathbf{y}\|
\end{aligned}$$

It then induces the conclusion. □

Lemma IV.2. *$\mathcal{L}_{Buzz} \downarrow$ when $S \downarrow$, for fixed G, ϕ, λ.*

Proof. Consider the simplest process when $S \downarrow$ by the definition of S, where one and only one partition S_w is divided into two pratitions S_{w_1}, S_{w_2}, such that $p(w)p(\mathbf{x}|w) = p(w_1)p(\mathbf{x}|w_1) + p(w_2)p(\mathbf{x}|w_2)$. The $\mathcal{L}_{Buzz}$ before and after dividing are denoted by Ω_1, Ω_2 respectively. The sign of $\Omega_1 - \Omega_2$ is decided by the change on $p(w)W^1[P(\mathbf{x}|w)\|P_G(\mathbf{y}|w)]\mathrm{d}w$.

Let $\gamma_1(\mathbf{x}, \mathbf{y})$ and $\gamma_2(\mathbf{x}, \mathbf{y})$ be the optimum couplings for $W^1[P(\mathbf{x}|w_1)\|P_G(\mathbf{y}|w_1)])$ and $W^1[P(\mathbf{x}|w_2)\|P_G(\mathbf{y}|w_2)]$. Let $\gamma(\mathbf{x}, \mathbf{y}) = \frac{1}{p(w)}(p(w_1)\gamma_1(\mathbf{x}, \mathbf{y}) + p(w_2)\gamma_2(\mathbf{x}, \mathbf{y}))$. Obviously, it is a coupling of $W^1[P(\mathbf{x}|w)\|P_G(\mathbf{y}|w)]$. It then induces the conclusion $\Omega_1 - \Omega_2 \geq 0$ by considering the minimality of $W^1[P(\mathbf{x}|w)\|P_G(\mathbf{y}|w)]$.

Each evolution from $S^{(n)}$ to $S^{(n-1)}$ can be divided into several such simplest processes. Then "$\Omega_1 \geq \Omega_2$" holds throughout the sequence, which implies $\mathcal{L}_{Buzz} \downarrow$ when $S \downarrow$. □

For simplicity, we denote $\max_{\phi,G,\lambda} \mathcal{L}_{Buzz}$ by $\max \mathcal{L}_{Buzz}$, and $\max_{\phi,G,\lambda} \mathcal{L}_{vae}$ by $\max \mathcal{L}_{Buzz}$. It is obvious that $\forall S$, $\mathrm{diag}\mathcal{X} = \{(\mathbf{x}, \mathbf{x}) \,|\, \mathbf{x} \in \mathcal{X}\} \subseteq S$. We shall soon discuss the limit case, $S \downarrow \mathrm{diag}\mathcal{X}$, *i.e.*, $p(\mathbf{x}|w)$ approaches a dirac distribution.

Lemma IV.3. *$\max \mathcal{L}_{Buzz} \geq \max \mathcal{L}_{vae}$. In addition, $\max \mathcal{L}_{Buzz} \downarrow \max \mathcal{L}_{vae}$ when $S \downarrow \mathrm{diag}\mathcal{X}$.*

Proof. Consider $\mathcal{L}_{vae}$ and $\mathcal{L}_{Buzz}$ with respect to the same ϕ, G, λ. Then $\gamma'(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}|w)\, p_G(\mathbf{y}|\mathbf{x})$ can be seen as a coupling of $(P(\mathbf{x}|w), P_G(\mathbf{y}|w))$.

$\mathbb{E}_{p(\mathbf{x}|w)} \mathbb{E}_{p_G(\mathbf{y}|\mathbf{x})} \|\mathbf{x} - \mathbf{y}\| \geq W^1[P(\mathbf{x}|w)\|P_G(\mathbf{y}|w)]$, which is obtained by the minimality of Wasserstein distance. It induces $\mathcal{L}_{Buzz} \geq \mathcal{L}_{vae}$, and further $\max \mathcal{L}_{Buzz} \geq \max \mathcal{L}_{vae}$.

When $S = \mathrm{diag}\mathcal{X}$, there is only one coupling, namely $p(\mathbf{x}|w)p_G(\mathbf{y}|\mathbf{x})$, for $W^1[P(\mathbf{x}|w)\|P_G(\mathbf{y}|w)]$, as [37] shows. It induces $\mathbb{E}_{p(\mathbf{x}|w)} \mathbb{E}_{p_G(\mathbf{y}|\mathbf{x})} - \|\mathbf{x} - \mathbf{y}\| = W^1[P(\mathbf{x}|w)\|P_G(\mathbf{y}|w)]$, and further, $\mathcal{L}_{Buzz} = \mathcal{L}_{vae}$. Therefore, $\max \mathcal{L}_{Buzz} = \max \mathcal{L}_{vae}$ when $S = \mathrm{diag}\mathcal{X}$. Recall Lemma IV.2 that $\max \mathcal{L}_{Buzz} \downarrow$ when $S \downarrow$, we can thus induce the conclusion. □

Lemma IV.4. *Let $p'_G(\mathbf{y}|\mathbf{x})$ denote $\mathbb{E}_{q_{\phi'}(\mathbf{z}|\mathbf{x})} [p_G(\mathbf{y}|\mathbf{z})]$. If (G, ϕ, λ) is a solution, then there exist (G, ϕ', λ), such that:*

$$\mathbb{E}_{p(\mathbf{x}|w)} \mathbb{E}_{p'_G(\mathbf{y}|\mathbf{x})} \|\mathbf{x} - \mathbf{y}\| = W^1[P(\mathbf{x}|w)\|P_G(\mathbf{y}|w)]$$

Then $\mathcal{L}_{Buzz} - \mathcal{L}'_{vae} = \mathcal{K}' - \mathcal{K}$ where $\mathcal{L}'_{vae}, \mathcal{K}'$ are defined with respect to the solution (G, ϕ', λ).

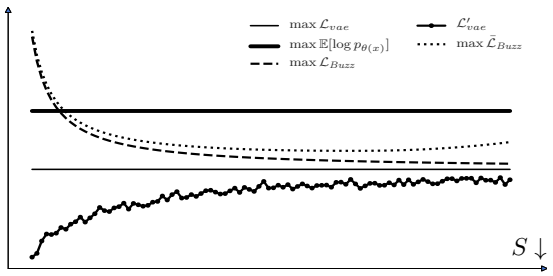


Fig. 5: Relationships of several losses in deduction. $\mathcal{L}'_{vae} \to \max \mathcal{L}_{vae}$ when $S \downarrow \mathrm{diag}\mathcal{X}$, which can be proved by combing Lemma IV.3 and Lemma IV.4. However, this tendency is not monotonic. $\max \bar{\mathcal{L}}_{Buzz}$ is the approximated loss to $\max \mathcal{L}_{Buzz}$.

Proof. Consider a simpler condition that $G(\mathbf{z})$ is injective, such that there exists an inverse function G^{-1} over the image of G. For a fixed w, let γ be the optimum coupling of $W^1[P(\mathbf{x}|w)\|P_G(\mathbf{y}|w)]$. Let $q_\gamma(\mathbf{y}|\mathbf{x}) = \frac{\gamma(\mathbf{x},\mathbf{y})}{p(\mathbf{x}|w)}$, $p_G(\mathbf{z}|\mathbf{y}) = \delta(\mathbf{z} - G^{-1}(\mathbf{y}))$. Let $q_{\phi'}(\mathbf{z}|\mathbf{x}) = \mathbb{E}_{q_\gamma(\mathbf{y}|\mathbf{x})} [p_G(\mathbf{z}|\mathbf{y})]$. Since $\gamma(\mathbf{x}, \mathbf{y})$ is optimum, we obtain:

$$\begin{aligned}
&p'_G(\mathbf{y}|\mathbf{x}) = \mathbb{E}_{q_{\phi'}(\mathbf{z}|\mathbf{x})} [p_G(\mathbf{y}|\mathbf{z})] = \int \mathbb{E}_{q_\gamma(\mathbf{y'}|\mathbf{x})} p_G(\mathbf{z}|\mathbf{y'}) p_G(\mathbf{y}|\mathbf{z}) \mathrm{d}\mathbf{z} \\
&= \mathbb{E}_{q_\gamma(\mathbf{y'}|\mathbf{x})} \left[\int p_G(\mathbf{y}|\mathbf{z}) p_G(\mathbf{z}|\mathbf{y'}) \mathrm{d}\mathbf{z} \right] \\
&= \mathbb{E}_{q_\gamma(\mathbf{y'}|\mathbf{x})} \left[\delta(\mathbf{y} - \mathbf{y'}) \right] = q_\gamma(\mathbf{y}|\mathbf{x}) = \frac{\gamma(\mathbf{x}, \mathbf{y})}{p(\mathbf{x}|w)}
\end{aligned}$$

It induces the conclusion:

$$\begin{aligned}
&\mathbb{E}_{p(\mathbf{x}|w)} \mathbb{E}_{p'_G(\mathbf{y}|\mathbf{x})} \|\mathbf{x} - \mathbf{y}\| = \iint p(\mathbf{x}|w) \frac{\gamma(\mathbf{x}, \mathbf{y})}{p(\mathbf{x}|w)} \|\mathbf{x} - \mathbf{y}\| \mathrm{d}\mathbf{y} \mathrm{d}\mathbf{x} \\
&= \iint \gamma(\mathbf{x}, \mathbf{y}) \|\mathbf{x} - \mathbf{y}\| \mathrm{d}\mathbf{y} \mathrm{d}\mathbf{x} = W^1[P(\mathbf{x}|w)\|P_G(\mathbf{y}|w)]
\end{aligned}$$

If G is not injective, the same conclusion still holds by setting $p_G(\mathbf{z}|\mathbf{y}) = \frac{p_G(\mathbf{y}|\mathbf{z}) p_\theta(\mathbf{z})}{\mathbb{E}_{p_\theta(\mathbf{z'})}[p_G(\mathbf{y}|\mathbf{z'})]}$, $q_{\phi'}(\mathbf{z}|\mathbf{x}) = \mathbb{E}_{q_\gamma(\mathbf{y}|\mathbf{x})} [p_G(\mathbf{z}|\mathbf{y})]$. Then repeat the above proof. Furthermore, by above equation, $p_G(\mathbf{y}|w) = p'_G(\mathbf{y}|w)$. Therefore, $\mathcal{L}_{Buzz} - \mathcal{L}'_{vae} = -\mathcal{K} + \mathcal{K}'$. □

For any solution ϕ, G, λ, a local optimal solution $(\bar{\phi}, G, \lambda)$ for $\mathcal{L}_{Buzz}$ is obtain by fixing $P_G(\mathbf{y}|w)$ and optimizing $\mathcal{K}$. $\bar{\phi} \sim \phi$ denotes $\bar{\phi} \in \{\phi' | p'_G(\mathbf{y}|w) = p_G(\mathbf{y}|w)\}$ (the *equivalent class* of ϕ). $\mathcal{L}_{Buzz}$ with respect to $(\bar{\phi}, G, \lambda)$ is:

$$-\lambda \mathbb{E}_{p(w)} W^1[P(\mathbf{x}|w)\|P_G(\mathbf{y}|w)] - \min_{\bar{\phi} \sim \phi} \bar{\mathcal{K}} - \log Z(\lambda)$$

From Lemma IV.4, we know $\exists \phi'$, $p_G(\mathbf{y}|w) = p'_G(\mathbf{y}|w)$, thus $\phi' \sim \bar{\phi} \sim \phi$, and the first term of the above equation can be replaced by another form. Then, $\mathcal{L}_{Buzz}$ with respect to $(\bar{\phi}, G, \lambda)$ is rewritten into another form $\mathcal{L}^\dagger_{Buzz}$ similar to $\mathcal{L}_{vae}$:

$$\mathcal{L}^\dagger_{Buzz} = \mathbb{E}_{p(\mathbf{x})} \left[\mathbb{E}_{q_{\phi'}(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) \right] - \min_{\bar{\phi} \sim \phi'} \bar{\mathcal{K}}$$

Because the optimum solution must also be a local optimal solution, $\max \mathcal{L}_{Buzz} = \max_{\phi',G,\lambda} \mathcal{L}^\dagger_{Buzz}$. It suggests that $\mathcal{L}_{Buzz}$ works by minimizing $\bar{\mathcal{K}}$ term on the equivalent class of ϕ'.

C. The Dual Form of $\mathcal{L}_{Buzz}$

Define Symbol $\mathcal{T}(f,w) = \mathbb{E}_{p(\mathbf{x}|w)} f(\mathbf{x}) - \mathbb{E}_{p_G(\mathbf{y}|w)} f(\mathbf{y})$. Symbol $\mathcal{T}^*(f,w)$ denotes $\mathcal{T}(R(f;S_w),w)$ where R is a functional for f, defined by $R(f;S_w)(\mathbf{x}) = f(\mathbf{x})$ for $\mathbf{x} \in S_w$ and $R(f;S_w)(\mathbf{y}) = \sup_{\mathbf{x}\in S_w}\{f(\mathbf{x}) - \|\mathbf{x}-\mathbf{y}\|\}$ for $\mathbf{y} \notin S_w$. In the absence of ambiguity, we use $\mathcal{T}(f)$, $\mathcal{T}^*(f)$ conveniently.

$Lip(f;\mathcal{S})$ denotes the minimum real constant $C \in \mathbb{R}$ such that $\|f(\mathbf{x}) - f(\mathbf{y})\| \le C\|\mathbf{x}-\mathbf{y}\|, \forall(\mathbf{x},\mathbf{y}) \in \mathcal{S}$. Let $\mathcal{Y}$ be the definition domain of function f, then $Lip(f)$ denotes $Lip(f;\mathcal{Y}\times\mathcal{Y})$. In particular, $f|_{S_w}$ denotes a function defined on S_w, then $Lip(f|_{S_w})$ denotes $Lip(f|_{S_w}; S_w \times S_w)$.

The *dual* form of $W^1[P(\mathbf{x}|w)\|P_G(\mathbf{y}|w)]$ on each partition relies on $Lip(f) \le 1$. It is intractable to find the function f for each partition in practice. Therefore, we need to reduce the search space for $\sup_{Lip(f)\le 1}\mathcal{T}(f)$ on an fixed partition.

Lemma IV.5. For a fixed w, define:

$$\mathcal{F} = \{f \big| Lip(f) \le 1\}, \mathcal{F}^* = \{f|_{S_w} \big| Lip(f|_{S_w}) \le 1\}$$

then $\sup_{f\in\mathcal{F}}\mathcal{T}(f) = \sup_{f|_{S_w}\in\mathcal{F}^*}\mathcal{T}^*(f|_{S_w})$.

Proof. Define $\kappa : \mathcal{F} \to \mathcal{F}^*$, a mapping limiting the definition domain of a given function f to the S_w. κ is surjective. We will show that $\sup_{\kappa^{-1}(f|_{S_w})}\mathcal{T}(f) = \mathcal{T}^*(f|_{S_w})$ for a fixed $f|_{S_w} \in \mathcal{F}^*$. $\mathcal{T}(f)$ can be decomposed as:

$$\mathbb{E}_{p(\mathbf{x}|w)} f(\mathbf{x}) - \int_{S_w} p_G(\mathbf{y}|w) f(\mathbf{y}) \mathrm{d}\mathbf{y} - \int_{\mathcal{X}\setminus S_w} p_G(\mathbf{y}|w) f(\mathbf{y}) \mathrm{d}\mathbf{y}$$

Consider $\sup_{\kappa^{-1}(f|_{S_w})}\mathcal{T}(f)$, whose values of 1st and 2nd term are fixed. In order to get a supremum, $f(\mathbf{y})$ need to be minimized in $\mathcal{X}\setminus S_w$. By $Lip(f) \le 1$, we get $f(\mathbf{y}) \ge \sup_{\mathbf{x}\in S_w}\{f(\mathbf{x}) - \|\mathbf{x}-\mathbf{y}\|\} = R(f|_{S_w}; S_w)$. By definition of R, and $f(\mathbf{y}) \ge R(f;S_w)(\mathbf{y})$, we get $\mathcal{T}(f) \le \mathcal{T}^*(f|_{S_w})$, so $\sup_{\kappa^{-1}(f|_{S_w})}\mathcal{T}(f) \le \mathcal{T}^*(f|_{S_w})$. We claim that $R(f|_{S_w}; S_w) \in \kappa^{-1}(f|_{S_w})$. We denote $R(f|_{S_w}; S_w)$ by f_R.

Obviously, $\kappa(f_R) = f|_{S_w}$. We only need to show $Lip(f_R) \le 1$. $\forall \mathbf{x} \in \bar{S}_w$ (the closure of S_w) and $\mathbf{y} \in \mathcal{X}$, $\|f_R(\mathbf{x}) - f_R(\mathbf{y})\| \le \|\mathbf{x}-\mathbf{y}\|$, which is obtained by the definition of $Lip(f|_{S_w}) \le 1$ and $f_R(\mathbf{y})$. Let $\mathbf{y}, \bar{\mathbf{y}} \in \mathcal{X}\setminus\bar{S}_w$, and without loss of generality, $f_R(\mathbf{y}) \ge f_R(\bar{\mathbf{y}})$. Let $\mathbf{x}^* \in \bar{S}_w$ be the optimal solution for $\sup_{\mathbf{x}\in S_w}\{f(\mathbf{x}) - \|\mathbf{x}-\mathbf{y}\|\}$. Since $\sup_{\mathbf{x}\in S_w}\{f(\mathbf{x}) - \|\mathbf{x}-\mathbf{y}\|\} = \sup_{\mathbf{x}\in\bar{S}_w}\{f_R(\mathbf{x}) - \|\mathbf{x}-\mathbf{y}\|\}$:

$$\begin{aligned} f_R(\mathbf{y}) - f_R(\bar{\mathbf{y}}) &= \sup_{\mathbf{x}\in S_w} f(\mathbf{x}) - \|\mathbf{x}-\mathbf{y}\| - \sup_{\mathbf{x}\in S_w} f(\mathbf{x}) - \|\mathbf{x}-\bar{\mathbf{y}}\| \\ &\le f_R(\mathbf{x}^*) - \|\mathbf{x}^*-\mathbf{y}\| - f_R(\mathbf{x}^*) + \|\mathbf{x}^*-\bar{\mathbf{y}}\| \le \|\mathbf{y}-\bar{\mathbf{y}}\| \end{aligned}$$

So $f_R \in \kappa^{-1}(f|_{S_w})$, and then $\sup_{\kappa^{-1}(f|_{S_w})}\mathcal{T}(f) \ge \mathcal{T}(f_R) = \mathcal{T}^*(f|_{S_w})$. It induces that $\sup_{\kappa^{-1}(f|_{S_w})}\mathcal{T}(f) = \mathcal{T}^*(f|_{S_w})$.

$$\sup_{\mathcal{F}}\mathcal{T}(f) = \sup_{f|_{S_w}\in\mathcal{F}^*}\sup_{\kappa^{-1}(f|_{S_w})}\mathcal{T}(f) = \sup_{f|_{S_w}\in\mathcal{F}^*}\mathcal{T}^*(f|_{S_w}) \square$$

Theorem IV.6. The *dual* form of $\mathcal{L}_{Buzz}$ is

$$\mathcal{L}_{Buzz} = -\lambda \sup_{Lip(F;S)\le 1} \mathbb{E}_{p(w)}\mathcal{T}^*(F) - \mathcal{K} - \log Z(\lambda)$$

Proof. Using the *dual* form of $W^1[P(\mathbf{x}|w)\|P_G(\mathbf{y}|w)]$ and Lemma IV.5, we obtain:

$$\begin{aligned} \mathcal{L}_{Buzz} &= -\lambda\,\mathbb{E}_{p(w)} \sup_{Lip(f)\le 1} \mathcal{T}(f) - \mathcal{K} - \log Z(\lambda) \\ &= -\lambda\,\mathbb{E}_{p(w)} \sup_{Lip(f|_{S_w})\le 1} \mathcal{T}^*(f|_{S_w}) - \mathcal{K} - \log Z(\lambda) \end{aligned}$$

We can obtain a function F defined on the whole space $\mathcal{X}$, composed from $f|_{S_w}$ over all S_w, by setting $F|_{S_w} = f|_{S_w}$, $\forall S_w$. This construction is denoted by $\mathcal{C}$. F is well-defined by the property of partitions. Define:

$$\mathcal{M}_F = \{F \big| Lip(F;S) \le 1\}$$

$$\mathcal{M}_f = \{M_f \big| M_f = \{f|_{S_w} \big| Lip(f|_{S_w}) \le 1\} \text{s.t.} \forall S_w \exists! f|_{S_w} \in M_f\}$$

$\mathcal{C}$ is a bijection between $\mathcal{M}_f$ and $\mathcal{M}_F$, since each $F \in \mathcal{M}_F$ can be constructed by exactly one M_f. It is obvious that $\mathbb{E}_{p(w)}\mathcal{T}^*(F) = \mathbb{E}_{p(w)}\mathcal{T}^*(f|_{S_w})$, where $F = \mathcal{C}(M_f)$. Thus,

$$\begin{aligned} \mathbb{E}_{p(w)} \sup_{Lip(f|_{S_w})\le 1} \mathcal{T}^*(f|_{S_w}) &= \sup_{M_f\in\mathcal{M}_f} \mathbb{E}_{p(w)}\mathcal{T}^*(f|_{S_w}) \\ &= \sup_{F\in\mathcal{M}_F} \mathbb{E}_{p(w)}\mathcal{T}^*(F) \end{aligned}$$

which induces the conclusion. □

D. From the Dual Form of $\mathcal{L}_{Buzz}$ to the Approximated $\tilde{\mathcal{L}}_{Buzz}$

In practice, it is hard to calculate $\mathcal{T}^*(F)$ since we don't know the exactly range of S_w but only the samples of $p(\mathbf{x}|w)$. Recall that in § III, we give a simple partition method: the partitions for space $\mathcal{X}$ is induced by the partitions on time. Each S_w is a connected component, so we could assume such partition to have good property, where $\int_{\mathcal{X}\setminus S_w} p_G(\mathbf{y}|w) f(\mathbf{y}) \mathrm{d}\mathbf{y}$ is almost 0. It further suggests it can be ignored, and thus a simple approximation approach is to replace $\mathcal{T}^*(F)$ by $\mathcal{T}(F)$.

Our approximation to the *dual* form of $\mathcal{L}_{Buzz}$ is given by:

$$\bar{\mathcal{L}}_{Buzz} = -\lambda \sup_{Lip(F;S)\le 1} \mathbb{E}_{p(w)}\mathcal{T}(F) - \mathcal{K} - \log Z(\lambda)$$

We use $|\mathcal{T}(F)|$ to replace $\mathcal{T}(F)$, to half the search space. By the gradient penalty [7], the limit of $Lip(f)$ can be achieved by a soft version of the constraint with a penalty on the gradient norm for random samples. Apply it for each partition. Define:

$$p(\hat{\mathbf{x}}) = \mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{p_G(\mathbf{y}|\mathbf{x})}\mathbb{E}_{\xi\sim[0,1]}\delta(\hat{\mathbf{x}} - (\xi\mathbf{x} + (1-\xi)\mathbf{y}))$$

We now obtain the *Buzz* training objective used in § III:

$$\begin{aligned} \tilde{\mathcal{L}}_{Buzz} = -\lambda \sup_F [\mathbb{E}_{p(w)}|\mathcal{T}(F)| - \eta\,\mathbb{E}_{p(\hat{\mathbf{x}})}(\|\nabla_{\hat{\mathbf{x}}}F(\hat{\mathbf{x}})\| - 1)^2] \\ -\mathcal{K} - \log Z(\lambda) \end{aligned}$$

where η is a hyper-parameter called gradient penalty weight. There is a limitation in our assumption that we use the exponential distribution as posterior distribution, instead of gaussian distribution which is used commonly but intractable for our derivation. We will work on it in the future.

V. Evaluation

A. Datasets

To evaluate *Buzz*, we obtain 11 well-maintained intricate KPIs from a large Internet company. These KPIs' time spans are long enough for training and evaluation. All KPIs have monitoring interval of 10 seconds between two observations.

We choose 3 datasets to investigate in detail, $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$, whose anomalies are thoroughly labeled by the operators that we work with. Although labels of the rest 8 KPIs are not as thorough, they represent more patterns of intricate KPIs. $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$, $\mathcal{E}$ and $\mathcal{F}$ are shown in Fig. 1.

Table I shows the detailed statistics of datasets $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and the average of all 11 KPIs. Since the inputs of our model are windows, we also count the number of total windows and abnormal windows, *i.e.*, the window contains at least one abnormal point. We divide each KPI into training, validation and testing sets, whose ratios are 56%, 14%, 30% respectively.

TABLE I: Statistics of $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and the average of all 11 KPIs.

DataSet	Total points	Anomaly points	Total windows	Abnormal windows
$\mathcal{A}$	172798	7352/4.25%	172671	33111/19.18%
$\mathcal{B}$	250460	2518/1.01%	250333	11911/4.76%
$\mathcal{C}$	259200	3512/1.35%	259073	21159/8.17%
Average	250550.73	2722.45/1.09%	250423.73	14091.81/5.63%

B. Performance

In our experiments, we set window size $W = 128$, dimension of z-space $K = 13$, gradient penalty weight $\eta = 10$, the number of critic iterations $n_{critic} = 3$, initial neighborhood size $s_0 = 32$ and initial batch size $b_0 = 8$. The parameters of Adam optimizer are $\alpha_0 = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$. We set $s \leftarrow s/2, b \leftarrow 2b$ after every 40 epochs. In detection part, repeat times T_{MC} is 10, and the sampling number L is 512.

In the variational and discriminative network, we use 4 convolutional layers, whose filters are all $(5, 5)$ and strides are $(2, 2), (1, 1), (2, 2), (1, 1)$ successively. The elements of $\boldsymbol{\epsilon}$ are all 0.0001. In the generative network, we use fully-connected layer to extract 512 features from $\mathbf{z}$, then we use 4 transposed convolutional layers to generate windows, whose filters and strides are the same as the variational network in inverse order.

We compare the performance of (unsupervised) *Buzz*, (unsupervised) *Buzz*-strict (defined in § III-F), Opprentice [3] (a state-of-art supervised approach that outperforms all traditional statistical models) and *Donut* [4] (a state-of-art unsupervised that outperforms Opprentice on smooth seasonal KPIs). Each model runs for 10 times on $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, and once on rest 8 datasets. The AUC and best F-score (two metrics used in *Donut*) of different approaches are shown in Fig. 6.

Next we show the training objective of *Buzz* on training set and validation set during training in Fig. 7. It shows that the adversarial training of *Buzz* is stable, and our simple approximation method is effective. Moreover, it intuitively shows that *Buzz* indeed maximizes ELBO.

Third, we compare $\mathrm{KL}\left[q_\phi(\mathbf{z}|\mathbf{x}) \,\middle\|\, p_\theta(\mathbf{z}|\mathbf{x})\right]$ of *Buzz* and *Donut*, which is calculated by the difference between log likelihood and ELBO, in Fig. 8. We calculate log likelihood by importance sampling [32] and ELBO over the normal windows in training set and testing set, respectively. It shows *Buzz* is well-trained and stable, while *Donut* is not on intricate KPIs.

Fourth, we show the mean performance of Opprentice, *Buzz*-strict, *Donut* and *Buzz* on all the 11 intricate KPIs in Fig. 9.

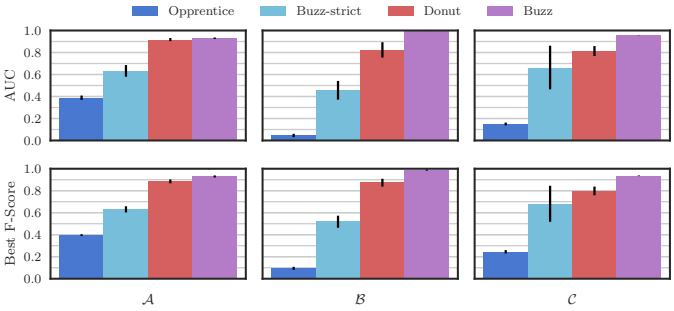


Fig. 6: The performance including AUC and best F-Score over $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$. Opprentice performs badly since there is no suitable traditional indicator for intricate KPIs. The performance of *Buzz*-strict is low with large variance, confirming our conjecture that $q_\phi(\mathbf{z}|\mathbf{x})$ deviates due to the influence of anomalies. The performance of *Donut* is usually good since it ignores the anomaly effect and uses MCMC imputation. The performance of *Buzz* is usually significantly better than *Donut* with less variance, because it considers the hard-training property of intricate KPIs and uses adversarial training to solve it.

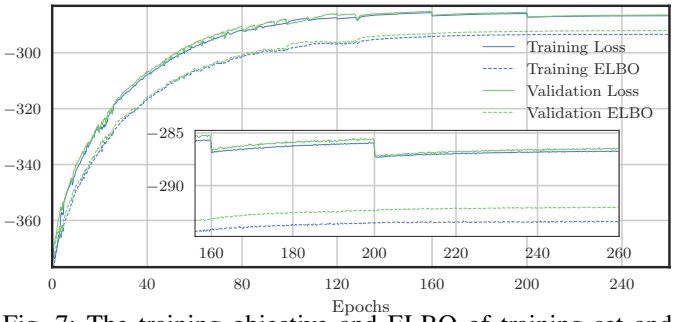


Fig. 7: The training objective and ELBO of training set and validation set during the training of *Buzz*. Before 160 epochs, generator and discriminator compete intensely, and the loss curve jitters. Jitters at 160, 200 epochs are caused by the changing of neighborhood size and $s = 1$ after 200 epochs. After 160 epochs, the training is stable. The losses at end of each 40 epochs represent the $\max \mathcal{L}_{Buzz}$ with enough training, for different s. The fact that $\max \mathcal{L}_{Buzz}$ decreases after 160 epochs, supports the theoretical analysis in Fig. 5. ELBO is calculated directly on solution ϕ, G, λ instead of ϕ' mentioned in § IV-B, which is intractable. It supports $\mathcal{L}_{Buzz} \geq \mathcal{L}_{vae}$ of Lemma IV.3. The fact that ELBO increases during the training, indicates our model maximizes the ELBO indeed.

Because thorough labels are only available on $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and Opprentice is a supervised algorithm, we only measure the performance of Opprentice on $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$. The results show that *Buzz* consistently works well on all 11 intricate KPIs.

VI. Conclusion

This paper proposes an adversarial training method in the Bayesian network based on partition analysis with solid theoretical proof. Based on it, we propose the first unsupervised anomaly detection algorithm *Buzz* for intricate KPIs with high performance. Its best F-scores on the data from a global Internet company range from 0.92 to 0.99, significantly outperforming existing approaches. We believe *Buzz*'s training method,

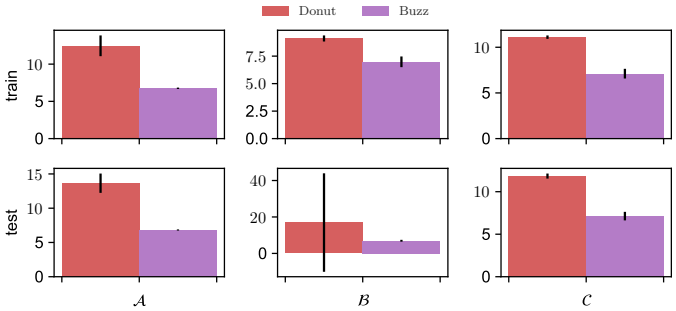


Fig. 8: KL $\left[q_\phi(\mathbf{z}|\mathbf{x}) \,\middle\|\, p_\theta(\mathbf{z}|\mathbf{x})\right]$ of *Donut* and *Buzz*. The $q_\phi(\mathbf{z}|\mathbf{x})$ is more similar to $p_\theta(\mathbf{z}|\mathbf{x})$ when the mean is less, and the training is more stable when the variance is less. *Donut* is unstable and overfittnig on $\mathcal{B}$ since its variance on testing set is much higher than training set. On average, the variance and mean of *Donut* are higher than *Buzz*. It shows *Buzz* is trained better. This confirms our conjecture in § II-D that our training method improves the training effect of VAE.

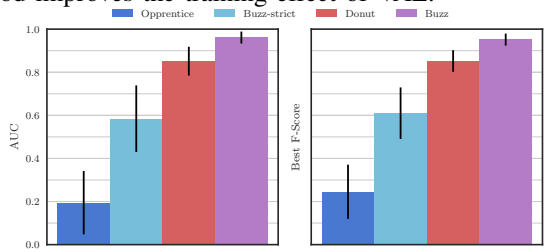


Fig. 9: The mean and variance of AUCs and best F-Scores over all 11 KPIs. Opprentice performs badly since there is no traditional indicator suitable for the intricate KPIs. On average, *Buzz* significantly outperforms the others.

detection method, and theoretical inference are significant first steps on tackling the training on data with intricate distribution and its anomaly detection. We plan to extend our work to many interesting and important directions: *e.g.*, an adversarial training method for common Gaussian posterior distribution; more advanced partition and approximation method.

VII. Acknowledgments

The authors gratefully acknowledge the contributions of Yijie Wu and Juexing Liao for their valuable suggestions. This work has been supported by the National Natural Science Foundation of China (NSFC) under grant 61472214, 61472210, the Beijing National Research Center for Information Science and Technology (BNRist) key projects, the Global Talent Recruitment (Youth) Program and Okawa Research Grant.

References

[1] S. Zhang, Y. Liu *et al.*, "Rapid and robust impact assessment of software changes in large internet-based services," in *CoNEXT*, 2015.
[2] ——, "Funnel: Assessing software changes in web-based services," *IEEE Transactions on Service Computing*, 2016.
[3] D. Liu, Y. Zhao *et al.*, "Opprentice: Towards practical and automatic anomaly detection through machine learning," in *IMC*, 2015.
[4] H. Xu, W. Chen *et al.*, "Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications," in *WWW*, 2018.
[5] J. Bu, Y. Liu *et al.*, "Rapid deployment of anomaly detection models for large number of emerging kpi streams," in *International Performance Computing and Communications Conference*. IEEE, 2018.
[6] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.
[7] I. Gulrajani, F. Ahmed *et al.*, "Improved training of wasserstein gans," in *NIPS*, 2017.
[8] S.-B. Lee, D. Pei *et al.*, "Threshold compression for 3g scalable monitoring," in *INFOCOM*. IEEE, 2012, pp. 1350–1358.
[9] B. Krishnamurthy, S. Sen *et al.*, "Sketch-based change detection: methods, evaluation, and applications," in *IMC*, 2003.
[10] A. H. Yaacob, I. K. Tan *et al.*, "Arima based network anomaly detection," in *ICCSN*. IEEE, 2010, pp. 205–209.
[11] H. Yan, A. Flavel *et al.*, "Argus: End-to-end service anomaly detection and localization from an isp's point of view," in *INFOCOM*. IEEE, 2012.
[12] Y. Chen, R. Mahajan *et al.*, "A provider-side view of web search response time," in *SIGCOMM*, ser. SIGCOMM '13, 2013.
[13] A. Mahimkar, Z. Ge *et al.*, "Rapid detection of maintenance induced changes in service performance," in *CoNEXT*. ACM, 2011.
[14] W. Lu and A. A. Ghorbani, "Network anomaly detection based on wavelet analysis," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, p. 4, 2009.
[15] N. Laptev, S. Amizadeh, and I. Flint, "Generic and scalable framework for automated time-series anomaly detection," in *SIGKDD*. ACM, 2015.
[16] Z. Fu, W. Hu, and T. Tan, "Similarity based vehicle trajectory clustering and anomaly detection," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 2. IEEE, 2005, pp. II–602.
[17] G. Münz, S. Li, and G. Carle, "Traffic anomaly detection using k-means clustering," in *GI/ITG Workshop MMBnet*, 2007.
[18] R. Laxhammar, G. Falkman, and E. Sviestins, "Anomaly detection in sea traffic-a comparison of the gaussian mixture model and the kernel density estimator," in *FUSION*. IEEE, 2009.
[19] M. Nicolau, J. McDermott *et al.*, "One-class classification for anomaly detection with kernel density estimation and genetic programming," in *EuroGP*. Springer, 2016.
[20] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," SNU Data Mining Center, Tech. Rep., 2015.
[21] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *ICLR*, 2014.
[22] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *ICML*, 2014.
[23] I. Goodfellow, J. Pouget-Abadie *et al.*, "Generative adversarial nets," in *NIPS*, 2014.
[24] A. Makhzani, J. Shlens *et al.*, "Adversarial autoencoders," *arXiv preprint arXiv:1511.05644*, 2015.
[25] I. Tolstikhin, O. Bousquet *et al.*, "Wasserstein auto-encoders," *stat*, vol. 1050, p. 19, 2017.
[26] T. Salimans, H. Zhang *et al.*, "Improving gans using optimal transport," *arXiv preprint arXiv:1803.05573*, 2018.
[27] A. B. L. Larsen, S. K. Sønderby *et al.*, "Autoencoding beyond pixels using a learned similarity metric," in *ICML*, 2016.
[28] L. Gyorfi and M. Kohler, "Nonparametric estimation of conditional distributions," *IEEE Transactions on Information Theory*, vol. 53, no. 5, p. 1872, 2007.
[29] C. A. Di Prisco, J. Llopis, and S. Todorcevic, "Borel partitions of products of finite sets and the ackermann function," *Journal of Combinatorial Theory, Series A*, vol. 93, no. 2, pp. 333–349, 2001.
[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2014.
[32] Y. Burda, R. Grosse, and R. Salakhutdinov, "Importance weighted autoencoders," *arXiv preprint arXiv:1509.00519*, 2015.
[33] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *ICLR*, 2014.
[34] S. T. Rachev *et al.*, "Duality theorems for kantorovich-rubinstein and wasserstein functionals," 1990.
[35] O. Bousquet, S. Gelly *et al.*, "From optimal transport to generative modeling: the vegan cookbook," *arXiv preprint arXiv:1705.07642*, 2017.
[36] S. Saks, "Theory of the integral," 1937.
[37] L. Guibas, D. Morozov, and Q. Mérigot, "Witnessed k-distance," *Discrete & Computational Geometry*, vol. 49, no. 1, pp. 22–45, 2013.