Quadratic Neuron-empowered Heterogeneous Autoencoder for Unsupervised Anomaly Detection

Jing-Xiao Liao, *Graduate Student Member, IEEE*, Bo-Jian Hou, Hang-Cheng Dong, Hao Zhang
Xiaoge Zhang, *Member, IEEE*, Jinwei Sun, Shiping Zhang, Feng-Lei Fan, *Member, IEEE*

***Abstract*—Inspired by the complexity and diversity of biological neurons, a quadratic neuron is proposed to replace the inner product in the current neuron with a simplified quadratic function. Employing such a novel type of neurons offers a new perspective on developing deep learning. When analyzing quadratic neurons, we find that there exists a function such that a heterogeneous network can approximate it well with a polynomial number of neurons but a purely conventional or quadratic network needs an exponential number of neurons to achieve the same level of error. Encouraged by this inspiring theoretical result on heterogeneous networks, we directly integrate conventional and quadratic neurons in an autoencoder to make a new type of heterogeneous autoencoders. To our best knowledge, it is the first heterogeneous autoencoder that is made of different types of neurons. Next, we apply the proposed heterogeneous autoencoder to unsupervised anomaly detection for tabular data and bearing fault signals. The anomaly detection faces difficulties such as data unknownness, anomaly feature heterogeneity, and feature unnoticeability, which is suitable for the proposed heterogeneous autoencoder. Its high feature representation ability can characterize a variety of anomaly data (heterogeneity), discriminate the anomaly from the normal (unnoticeability), and accurately learn the distribution of normal samples (unknownness). Experiments show that heterogeneous autoencoders perform competitively compared to other state-of-the-art models.**

***Impact Statement*—In this work, we present a novel perspective on neural network design by introducing neuronal diversity. Firstly, we prove that neural networks that combine two types of neurons have superior theoretical approximation efficiency compared to networks comprising solely homogeneous neurons. Secondly, we develop the first autoencoder that is made of different neurons, which is a new addition to the family of autoencoders. Lastly, the proposed heterogeneous autoencoder performs better than the state-of-the-art models in tabular and bearing fault unsupervised anomaly detection. In brief, both theoretical derivation and experimental results confirm the benefit of neuronal diversity in neural network design.**

***Index Terms*—Deep learning theory, heterogeneous autoencoder, quadratic neuron, anomaly detection.**

The work described in this paper was partially supported by the Direct Grant for Research from the Chinese University of Hong Kong and ITS/173/22FP from the Innovation and Technology Fund of Hong Kong, was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU 25206422), and was partially supported by the Research Committee of The Hong Kong Polytechnic University under project code G-UAMR and student account code RL3C. *(Corresponding Authors: Shiping Zhang, Feng-Lei Fan)*

Jing-Xiao Liao is with School of Instrumentation Science and Engineering, Harbin Institute of Technology, Harbin, China, and is also with Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong, SAR of China

Hang-Cheng Dong, Jinwei Sun, Shiping Zhang are with School of Instrumentation Science and Engineering, Harbin Institute of Technology, Harbin, China. (email: spzhang@hit.edu.cn)

Bo-Jian Hou, Hao Zhang are with Weill Cornell Medicine, Cornell University, New York City, US.

Xiaoge Zhang is with Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong, SAR of China.

Feng-Lei Fan is with Department of Mathematics, The Chinese University of Hong Kong, Hong Kong, SAR of China. (email: hitfanfenglei@gmail.com)

I. Introduction

DEEP learning has achieved great success in many important fields [1]. However, so far, deep learning innovations mainly revolve around structure design such as increasing depth and proposing novel shortcut architectures [2–4]. Almost exclusively, the existing advanced networks only involve the same type of neurons characterized by i) the inner product of the input and the weight vector; ii) a nonlinear activation function. For convenience, we call this type of neurons *conventional neurons*. Although these neurons can be interconnected to approximate a general function, we argue that the type of neurons useful for deep learning should not be unique. As we know, neurons in a biological neural system are diverse, collaboratively generating all kinds of intellectual behaviors. Since a neural network was initially devised to imitate a biological neural system, neuronal diversity should be carefully examined in neural network research.

In this context, a new type of neurons called *quadratic neurons* was recently proposed in [5], which replace the inner product in a conventional neuron with a simplified quadratic function. Hereafter, we call a network made of quadratic neurons a *quadratic network* and a network made of conventional neurons a *conventional network*. The superiority of quadratic networks over conventional networks in representation efficiency and capacity was demonstrated in [6]. Intuitively, nonlinear features ubiquitously exist in real world, and quadratic neurons are empowered with a quadratic function to represent nonlinear features efficiently and effectively. Mathematically, when the rectified linear unit (ReLU) is employed, a conventional model is a piecewise linear function, whereas a quadratic one is a more powerful piecewise nonlinear mapping. Since a quadratic neuron shows great potential, we are encouraged to keep pushing its theory and application boundaries.

Previously, it was shown that there exists a function such that a conventional network needs an exponential number of neurons to approximate, but a quadratic network only needs a polynomial number of neurons to achieve the same level of error [7]. Despite the attraction of this construction, with the help of techniques in [8], we find that one can also construct a function that can realize the opposite situation, *i.e.,* a function that is hard to approximate by a quadratic

network but easy to approximate by a conventional network (Theorem 2). Combining these two constructed functions, one can naturally get a function easy to approximate by a heterogeneous network of both quadratic and conventional neurons but hard by a purely conventional or quadratic network, where the difference remains polynomial vs exponential (Theorem 1). Moreover, the opposite case will not happen because the purely conventional or quadratic is a degenerated case of the heterogeneous network. Such a theoretical derivation makes the employment of heterogeneous models well grounded, implicating that there are at least some tasks pretty suitable for the combination of conventional and quadratic neurons so that an exponential difference is reached. While for the general task, heterogeneous networks' performance will not be worse, since a purely conventional or quadratic network is the special case of a heterogeneous network.

This theoretical derivation agrees with our intuition. There is no one-size-fits-all artificial neuron in deep learning. Both conventional and quadratic neurons cannot perform well on all kinds of tasks. Encouraged by the inspiring theoretical result on heterogeneous networks and the idea of neuronal synergy, here we directly integrate conventional and quadratic neurons in an autoencoder to make a new type of *heterogeneous autoencoders*. Autoencoders [5; 9–11] are widely used in unsupervised learning tasks such as feature extraction [12], denoising[13], anomaly detection [14], and so on. To the best of our knowledge, all the previous mainstream autoencoders use the same type of neurons regardless of their innovations. The proposed heterogeneous autoencoder is the first heterogeneous autoencoder made up of diverse neurons in the family of autoencoders. Furthermore, we apply the heterogeneous autoencoder to solve the anomaly detection problem.

Anomaly detection is widely applied in different contexts such as intelligent manufacturing, data management, and intelligent transportation [15; 16]. However, anomaly detection is challenging due to the following issues [17]:

- **Unknownness.** The anomaly features may include the unknown data, which requires an anomaly detection model to accurately construct the distribution of normal samples and be sensitive to anomalous samples.
- **Heterogeneity.** Anomalies are irregular, and different anomalies may display completely different abnormal features. For instance, when detecting network traffic attacks, four types of attacks are all anomalies (DOS, R2L, U2R, Probing).
- **Unnoticeability.** Anomalies are typically rare and are easily overwhelmed by normal instances. Therefore, it is difficult to detect the anomaly.

Based on the above analyses, unsupervised anomaly detection highly relies on the models' ability to feature representation. A model with a high feature representation ability can characterize a variety of anomaly data (heterogeneity), discriminate the anomaly from the normal (unnoticeability), and accurately learn the distribution of normal samples (unknownness). Our theory indicates that a heterogeneous network has a more powerful representation ability than a homogeneous network; therefore, a heterogeneous network is suitable to address the anomaly detection problem. Experiments demonstrate that a heterogeneous autoencoder achieves competitive performance relative to other state-of-the-art on anomaly detection datasets and real-world bearing fault detection. In summary, our contributions are

1) We prove that to approximate a constructed function, a heterogeneous network is more efficient and has a more powerful capability than a homogeneous one. Different from ensemble learning, our result characterizes the ability of heterogeneous networks from the perspective of approximation theory, which is a novel theoretical angle and paves a way for the employment of heterogeneous networks.
2) To further verify our theory, we develop a first-of-its-kind autoencoder that integrates essentially different types of neurons in one model, which is a methodological contribution to autoencoder design.
3) Experiments suggest that the proposed heterogeneous autoencoder delivers competitive performance compared to homogeneous autoencoders and other baselines on unsupervised tabular data anomaly detection.

II. Related Works

Anomaly Detection (AD) algorithms strive to identify outliers that deviate significantly from the majority of data [18]. This study concentrates on unsupervised AD, which exclusively utilizes samples from the normal class to construct a model for outlier detection [19]. Broadly, AD algorithms can be categorized into traditional machine learning (shallow) and deep learning-based (deep) methods [20]. First, traditional machine learning methods are roughly summarised as follows: i) Classification-based method, one-class support vector machine (OCSVM) [21] and support vector data description (SVDD) [22]; ii) Probabilistic-based method, Gaussian mixture model (GMM) [23] and kernel density estimation (KDE) [24]; iii) Reconstruction-based method, probabilistic principal component analysis (p-PCA) [25]; iv) Distance-based method, local outlier factor (LOF) [26] and empirical cumulative outlier detection (ECOD) [27].

On the other hand, deep learning-based methods that are capable of handling larger and high-dimensional data have recently garnered increased attention. Compared to machine learning methods, deep neural network methods demonstrate the superior generalizability to various types of data. Several approaches integrated deep neural networks into machine learning methods, such as DeepSVDD [28], deep autoencoding GMM (DAGMM) [29], and deep autoencoding support vector data descriptor (DASVDD) [19]. These approaches are predicated on certain prior assumptions. Conversely, other approaches adopted pure deep neural networks to construct the end-to-end AD models without requiring any prior knowledge. The primary objective of these approaches is to enhance the performance and generalizability of AD methods across a broad spectrum of data. The most representative methods are autoencoders (AEs) [30; 31] and generative adversarial networks (GANs) [32; 33], which automatically learn the latent features of normal samples to enlarge their differences from the outliers in the latent space.

This paper specifically focuses on autoencoders for anomaly detection. Compared to GANs, AEs can primarily be adopted for anomaly detection and exhibit lower computational complexity. This also facilitates the easier integration of different neurons.

Autoencoders (AEs) constitute a vibrant research field in machine learning and have attained significant success in a variety of tasks, including data generation [34; 35], fault diagnosis [10; 12], and anomaly detection [30; 31]. Presently, autoencoders, as a class of reconstruction-based nonlinear dimension reduction models, have emerged as the most extensively adopted methods for unsupervised anomaly detection, owing to their lightweight and user-friendly variants [20]. The advantages of autoencoders, such as automatic latent feature extraction, reduction of dimensional curses, and potent nonlinear representation ability, have been proven beneficial in addressing the challenges associated with anomaly detection. Recent developments in autoencoders can be categorized into two aspects. i) Ensemble learning has been adapted to the autoencoder: Pan *et al.* [36] proposed a synchronized heterogeneous autoencoder that uses an item-oriented and a user-oriented autoencoder to serve as a teacher and a student, respectively, towards the distillation of both label-level and feature-level knowledge. Some studies combine multiple autoencoders for heterogeneous data, *e.g.*, combining a denoising autoencoder and a recurrent autoencoder for non-sequential and sequential data, respectively [37]. ii) Some novel metrics to measure the reconstruction error: Hojjati *et al.* [19] proposed an anomaly score that combines the reconstruction error and the distance of the enclosing hypersphere in the latent representation, and constructed an autoencoder called DASVDD. The low-rank and sparse priors were integrated into the loss function of deep autoencoders and employed in the hyperspectral anomaly detection [38].

However, regardless of representation regularization, architecture modification, and direct combination, these autoencoders are built upon the same type of neurons, which are conventional neurons. In contrast, the proposed heterogeneous autoencoders design an autoencoder by replacing neuron types, which provides a fresh perspective on autoencoder design.

Heterogeneous neural networks, characterized by the integration of various sub-networks or models, have shown significant potential in tackling complex scenarios such as big data and multi-modality. Several notable works include the heterogeneous Rybak neural network (HRYNN) introduced by Qi et al., which is composed of multiple Rybak neural networks (RYNNs) has demonstrated excellent image enhancement effects, particularly on the image edge details [39]. Sadr et al. constructed heterogeneous neural networks by integrating intermediate features from convolutional and recursive neural networks, exhibiting superior efficiency and generalization performance in sentiment analysis [40]. Another example is a heterogeneous neural network that integrates a hierarchical fused neural fuzzy and neural network, providing effective representation learning of users and items for multiple recommendation problems [41].

On the other hand, some multimodal large language models (MLLMs), despite not being explicitly named "heterogeneous networks", can be broadly considered as a variant of heterogeneous neural network [42]. These models utilize different sub-networks to handle data of different modalities. For instance, contrastive language-image pre-training (CLIP) [43] and OpenCLIP [44] construct both image and text encoders to train image-text pairs. These large multimodal models have shown powerful performance in various downstream tasks, such as zero-shot classification, retrieval, linear probing, and end-to-end fine-tuning. The concept of heterogeneous networks is also implied in some text-to-image diffusion models. For example, ControlNet injects additional conditions into a neural network block and incorporates it as a branch of the large pre-trained stable diffusion model [45] for fine-tuning [46]. Another text-conditional image diffusion model, RAPHAEL, is capable of generating highly artistic images through text descriptions. It stacks space and time mixture-of-experts (MoEs) layers, which consist of distinguished neural structures and construct billions of diffusion paths from the input to the output [47].

In summary, the principle of heterogeneity is prevalent in the design of deep learning networks. Integrating diverse architecture is beneficial in handling complex tasks. This characteristic is also advantageous for unsupervised AD, which requires a strong representation ability to represent distinguished latent features. Our heterogeneous network is different from the above in terms of that the heterogeneity lies in the incorporation of two kinds of neurons, providing superior representation of both linear and quadratic features.

Polynomial networks. Recently, higher-order units were revisited in the context of deep learning [48; 49]. 1) Chrysos *et al.* [49; 50] successfully applied the high-order units into a deep network by reducing the complexity of the individual high-order unit via tensor decomposition. They investigated the architecture of polynomial networks and achieved state-of-the-art results in image classification [51], as well as its robustness property [52]. To achieve the parametric efficiency, a plethora of polynomial neuron studies focused on quadratic neurons. Table I outlines the latest-proposed quadratic neurons. The complexity of neurons in [48; 53–55] is of $\mathcal{O}(n^2)$, but the quadratic neuron we use has only $\mathcal{O}(3n)$ parameters and scales well when a network becomes deep. The neurons in [49; 56–58] are a special case of [5], which means that this neuron is expressive. Therefore, we think that the quadratic neuron we use achieves an excellent balance between scalability and expressivity. 2) Neurons with polynomial activation [59] can also lead to a polynomial network. However, some experiments [6] show that using polynomial activation often causes a network not to converge and cannot obtain meaningful results.

There exist non-linear CNN models that use bilinear or trilinear attention modules, which are popular in image classification and reconstruction tasks [61; 62]. These models enhance their representation abilities by devising novel non-linear attention modules or layers. The equations for these attention modules resemble polynomial neurons, such as $\boldsymbol{x}^\top \boldsymbol{W}\boldsymbol{x}$ [63] or $((\boldsymbol{W}\boldsymbol{x}^\top)\boldsymbol{x})\boldsymbol{x}$ [64]. For example, Sun et al. proposed a higher-order polynomial transformer (HP-Transformer) for

TABLE I: A summary of the latest-proposed quadratic neurons. $\sigma(\cdot)$ denotes the nonlinear activation function. $\odot$ denotes Hadamard product. Here, $\boldsymbol{W} \in \mathbb{R}^{n\times n}$, $\boldsymbol{w}, \boldsymbol{w}^r, \boldsymbol{w}^g, \boldsymbol{w}^b \in \mathbb{R}^{n\times 1}$, and we omit the bias terms in these neurons.

Papers	Formulations
Zoumpourlis *et al.*(2017) [48; 53]	$\sigma(\boldsymbol{x}^\top \boldsymbol{W}\boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{w})$
Jiang *et al.*(2019) [54] Mantini&Shah(2021) [55]	$\sigma(\boldsymbol{x}^\top \boldsymbol{W}\boldsymbol{x})$
Goyal *et al.*(2020) [56]	$\sigma((\boldsymbol{x}\odot\boldsymbol{x})^\top \boldsymbol{w})$
Bu&Karpatne(2021) [57]	$\sigma((\boldsymbol{x}^\top \boldsymbol{w}^r)\cdot(\boldsymbol{x}^\top \boldsymbol{w}^g))$
Xu *et al.*(2022) [58]	$\sigma((\boldsymbol{x}^\top \boldsymbol{w}^r)\cdot(\boldsymbol{x}^\top \boldsymbol{w}^g) + \boldsymbol{x}^\top \boldsymbol{w}^b)$
Fan *et al.*(2018) [60]	$\sigma((\boldsymbol{x}^\top \boldsymbol{w}^r)(\boldsymbol{x}^\top \boldsymbol{w}^g) + (\boldsymbol{x}\odot\boldsymbol{x})^\top \boldsymbol{w}^b)$

fine-grained freezing of gait patterns [65]. HP-Transformer focuses on high-order gait patterns. However, they differ fundamentally from polynomial neurons. The attention modules implement non-linear operations through the sum and product of convolution layers, which provide a global attention mechanism. Polynomial neurons are point-wise and provide a local self-attention mechanism [66]. This local attention is efficient and lightweight in extracting latent features from tabular datasets compared to attention modules. Therefore, we develop quadratic neurons for anomaly detection tasks.

This work is essentially different from our previous studies. In [5], we introduced a quadratic convolutional autoencoder to address the issue of denoising in low-dose computed tomography. [6] highlighted the expressivity of quadratic networks by using the spline theory and a measure from algebraic geometry, and presented an effective training method called ReLinear. [7] established a theorem that demonstrates the superior approximation capability of quadratic networks compared to conventional networks in some functions, while the main theorem in this draft provides a theory for heterogeneous networks. These earlier studies did not touch on the idea of synergizing different neurons into a network and the efficiency of heterogeneous networks.

III. Power of Heterogeneous Networks

Here, we prove a non-trivial theorem that there exists a function that is easy to approximate by a heterogeneous network but hard to approximate by a purely conventional or quadratic network, *i.e.*, the difference in the number of neurons needed for the same level of error is polynomial vs exponential. This result suggests that at least for some tasks, combining conventional and quadratic neurons is instrumental to model's power and efficiency, which makes the employment of heterogeneous networks well-supported.

Quadratic neuron [5]. A quadratic neuron computes two inner products and one power term of the input vector and aggregates them for a nonlinear activation function. Mathematically, given the d-dimension input $\boldsymbol{x} = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^{d\times 1}$, the output of a quadratic neuron is expressed as

$$\begin{aligned}&\sigma\Big((\sum_{i=1}^{d} w_i^r x_i + b^r)(\sum_{i=1}^{d} w_i^g x_i + b^g) + \sum_{i=1}^{d} w_i^b x_i^2 + c\Big)\\ =&\sigma\Big((\boldsymbol{x}^\top \boldsymbol{w}^r + b^r)(\boldsymbol{x}^\top \boldsymbol{w}^g + b^g) + (\boldsymbol{x}\odot\boldsymbol{x})^\top \boldsymbol{w}^b + c\Big),\end{aligned} \tag{1}$$

where $\sigma(\cdot)$ is a nonlinear activation function, $\odot$ denotes the Hadamard product, $\boldsymbol{w}^r, \boldsymbol{w}^g, \boldsymbol{w}^b \in \mathbb{R}^{d\times 1}$ are weight vectors, and $b^r, b^g, c \in \mathbb{R}$ are biases. Please note that superscripts r, g, b are just marks for convenience without special meanings.

TABLE II: A table of important symbols

Symbol	Description
$\boldsymbol{x}$	Input
d	The dimension of the input
$\boldsymbol{w}^r$, $\boldsymbol{w}^g$, $\boldsymbol{w}^b$	Weight vectors in a quadratic neuron
b^r, b^g, c	Biases
σ	The activation function
$\boldsymbol{\omega}$	Frequency elements
$\boldsymbol{v}$	Unit vector
$\mathbb{B}_d$	Unit Euclidean ball
$R_d\mathbb{B}_d$	Unit-volume Euclidean ball, R_d is a selected constant.
$\mathbf{1}_{\boldsymbol{\omega}\in R_d\mathbb{B}_d}$	Indicator function that elements within $R_d\mathbb{B}_d$ are one
$\varphi(\boldsymbol{x})$	Fourier transform of $\mathbf{1}_{\boldsymbol{\omega}\in R_d\mathbb{B}_d}$
$\mu = \varphi^2(\boldsymbol{x})$	A general measure
$\mathbb{E}_{\boldsymbol{x}\sim\mu}(p-q)^2$	The distance between p and q under the measure μ
f_1	A one-hidden-layer conventional network
f_2	A one-hidden-layer quadratic network
f_3	A one-hidden-layer heterogeneous network
$\tilde{g}_{ip}(\boldsymbol{x})$	The constructed inner-product function
$\tilde{g}_r(\boldsymbol{x})$	The constructed radial function
c_σ	The constant dependent on σ
c_1, c_2	Constants associated with the quadratic network
ϵ_1	Approximation error associated with quadratic networks
c_3, c_4	Constants associated with the conventional network
ϵ_2	Approximation error associated with conventional networks
C_1, C_2	Constants associated with heterogeneous networks
$\mathrm{Span}\{\boldsymbol{v}\}$	The set of all linear combinations of $\boldsymbol{v}$
$\mathrm{Span}\{\boldsymbol{v}\} + R_d\mathbb{B}_d$	The region resultant by convoluting $\mathrm{Span}\{\boldsymbol{v}\}$ and $R_d\mathbb{B}_d$
$m_i(\boldsymbol{x})$	A continuous radial function
$\mathbb{S}^{d-1}$	The unit Euclidean sphere in $\mathbb{R}^d$
T	$T = \mathrm{Span}\{\boldsymbol{v}\} + R_d\mathbb{B}_d$
r	The radius
$h(r)$	$\frac{Area\left(r\mathbb{S}^{d-1}\cap T\right)}{Area\left(r\mathbb{S}^{d-1}\right)}$
c	A constant that restricts the upper bound of $h(r)$
l	The function $l = \frac{\widehat{\tilde{g}_{ip}\varphi}}{\left\|\tilde{g}_{ip}\varphi\right\|_{L_2}}$
q	The function $q = \frac{\widehat{f_2\varphi}}{\left\|f_2\varphi\right\|_{L_2}}$
$\tilde{g}_{ip}^{(t)}(\boldsymbol{x})$	Truncating $\tilde{g}_{ip}(\boldsymbol{x})$

Preliminaries and assumptions. Symbols used in our theorem are shown in Table II. Given arbitrary functions p and q, we have the following assumptions. i) $\|\cdot\|$ is the L_2 norm, *i.e.*, $\|p\|^2 = \int_{\boldsymbol{x}} p^2(\boldsymbol{x})d\boldsymbol{x}$. ii) The inner product of two functions is $\langle p, q\rangle_{L_2} = \int_{\boldsymbol{x}} p(\boldsymbol{x})q(\boldsymbol{x})d\boldsymbol{x}$. iii) $\varphi(\boldsymbol{x}) = (\frac{R_d}{\|\boldsymbol{x}\|})^{d/2} J_{d/2}(2\pi R_d\|\boldsymbol{x}\|)$, where $J_{d/2}: \mathbb{R}\to\mathbb{R}$ is the Bessel function of the first kind. $\varphi(\boldsymbol{x})$ is the Fourier transform of the unit-volume ball in the frequency domain $\mathbf{1}_{\boldsymbol{\omega}\in R_d\mathbb{B}_d}$. iv) For simplicity, we use $\hat{p}(\boldsymbol{\omega})$ to denote the Fourier transform of $p(\boldsymbol{x})$. v) The L_2-norm of $p(\boldsymbol{x})$ under a general measure $\mu = \varphi^2(\boldsymbol{x})$ is $\left[\int_{\boldsymbol{x}} p^2(\boldsymbol{x})\varphi^2(\boldsymbol{x})d\boldsymbol{x}\right]^{1/2}$. The distance between p and q under a general measure $\mu = \varphi^2(\boldsymbol{x})$ equals to

$$\int (p-q)^2\varphi^2 d\boldsymbol{x} = \|p\varphi - q\varphi\|^2 = \|\hat{p} * \mathbf{1}_{\boldsymbol{\omega}\in R_d\mathbb{B}_d} - \hat{q} * \mathbf{1}_{\boldsymbol{\omega}\in R_d\mathbb{B}_d}\|^2, \tag{2}$$

where $*$ is a convolution. For simplicity, let $\mathbb{E}_{\boldsymbol{x}\sim\mu}(p-q)^2 = \int (p-q)^2\varphi^2 d\boldsymbol{x}$. vi) A one-hidden-layer conventional network of k neurons is $f_1(\boldsymbol{x}) = \sum_{i=1}^{k} a_i\sigma(b_i\boldsymbol{x}^\top\boldsymbol{v}_i + c_i)$, while a one-hidden-layer quadratic network is simplified as $f_2(\boldsymbol{x}) = \sum_{i=1}^{k} a_i\sigma(b_i\boldsymbol{x}^\top\boldsymbol{x} + c_i)$, in which keeping only the power term is sufficient to express the construction. vii) All theorems

assume that the activation function $\sigma(\cdot)$ satisfies $|\sigma(x)| \leq C(1+|x|^{\alpha}), x \in \mathbb{R}$ for some constants C and α, where C and α are dependent on $\sigma(\cdot)$. viii) $\mathrm{Span}\{\boldsymbol{v}\}$ is the set of all linear combinations of $\boldsymbol{v}$. ix) $\mathrm{Span}\{\boldsymbol{v}\}+R_d\mathbb{B}_d$ is the region resultant from convolution between two regions: $\mathrm{Span}\{\boldsymbol{v}\}$ and $R_d\mathbb{B}_d$. Geometrically, $\mathrm{Span}\{\boldsymbol{v}\}+R_d\mathbb{B}_d$ is a hyper-cylinder. As Figure 1 shows, Fourier transform of a radial function is also radial in the frequency domain, while Fourier transform of an inner-product based function $g(\boldsymbol{x}^\top\boldsymbol{v})$ is supported over a line, denoted as $\mathrm{Span}\{\boldsymbol{v}\}$. Then, $\hat{g}*\mathbf{1}_{\boldsymbol{\omega}\in R_d\mathbb{B}_d}$ will be supported over a region resultant from convolution between two regions: $\mathrm{Span}\{\boldsymbol{v}\}$ and $R_d\mathbb{B}_d$. Geometrically, $\mathrm{Span}\{\boldsymbol{v}\}+R_d\mathbb{B}_d$ is a hyper-cylinder.

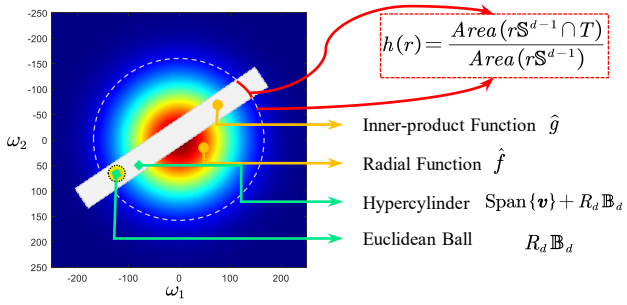


Fig. 1: Fourier transforms of a radial function f and an inner-product based function g in the frequency domain.

Theorem 1 (Main)**.** *There exist universal constants c_σ, $c_1,c_2,c_3,c_4,C_1,C_2,\epsilon_1,\epsilon_2>0$ such that the following holds: For every dimension d, there exists a measure μ and a function $\tilde{g}_{ip}(\boldsymbol{x})+\tilde{g}_r(\boldsymbol{x}):\mathbb{R}^d\to\mathbb{R}$ with the following properties:*

1. *Every f_1 expressed by a one-hidden-layer conventional network of width at most $\frac{1}{2}c_1e^{c_2d}$ has*
$$\mathbb{E}_{\boldsymbol{x}\sim\mu}[f_1-(\tilde{g}_{ip}+\tilde{g}_r)]^2\geq\epsilon_2. \tag{3}$$
2. *Every f_2 expressed by a one-hidden-layer quadratic network of width at most $\frac{1}{2}c_3e^{c_4d}$ has*
$$\mathbb{E}_{\boldsymbol{x}\sim\mu}[f_2-(\tilde{g}_{ip}+\tilde{g}_r)]^2\geq\epsilon_1. \tag{4}$$
3. *There exists a function f_3 expressed by a one-hidden-layer heterogeneous network that can approximate $\tilde{g}_{ip}+\tilde{g}_r$ with $C_1c_\sigma d^{3.75}+C_2c_\sigma$ neurons.*

The sketch of proof (Table III). It has been shown that there exists a function $\tilde{g}_r$ hard to approximate by a one-hidden-layer conventional network and easy to approximate by a one-hidden-layer quadratic network (Theorem 3). We only need to construct a function $\tilde{g}_{ip}$ hard to approximate by a one-hidden-layer quadratic network and easy to approximate by a one-hidden-layer conventional network (Theorem 2). Then, $\tilde{g}_{ip}+\tilde{g}_r$ will be hard for both conventional and quadratic networks but easy for a heterogeneous network (Theorem 1).

Remark 1. The standard quadratic neuron is $\sigma((\boldsymbol{x}^\top\boldsymbol{w}^r+b^r)(\boldsymbol{x}^\top\boldsymbol{w}^g+b^g)+(\boldsymbol{x}\odot\boldsymbol{x})^\top\boldsymbol{w}^b+c)$. By excluding interaction terms, we do not intend to weaken the expressive ability of quadratic neurons to establish the superiority of heterogeneous networks. Instead, this is because the target function $\tilde{g}_{ip}$ is a function made of an inner product. Our theorem is existential. Even if we include interaction terms, to represent $\tilde{g}_{ip}$, we can find a network whose every quadratic neuron actually takes $\boldsymbol{w}^g=0,b^g=1,\boldsymbol{w}^b=0,c=0$. Such a network is essentially a conventional network, with quadratic neurons degenerating into conventional ones. In this regard, it is true that a standard quadratic network can express $\tilde{g}_{ip}+\tilde{g}_r$. But because some quadratic neurons degenerate into conventional neurons, this quadratic network is essentially a heterogeneous network. If the constructed function $\tilde{g}_{ip}$ is not based on inner-product, the original form of quadratic neurons must be adopted.

TABLE III: The approximability of $\tilde{g}_{ip}$, $\tilde{g}_r$, and $\tilde{g}_{ip}+\tilde{g}_r$.

	construction	conventional f_1	quadratic f_2	heterogeneous f_1+f_2
Theorem 3	$\tilde{g}_r$ (radial)	✗	✓	-
Theorem 2	$\tilde{g}_{ip}$ (inner-product)	✓	✗	-
Theorem 1	$\tilde{g}_{ip}+\tilde{g}_r$	✗	✗	✓

Theorem 2. *There exist universal constants c_σ, $c_1,c_2,C_1,\epsilon_1>0$ such that the following holds: For every dimension d, there exists a measure μ and an inner-product based function $\tilde{g}_{ip}(\boldsymbol{x}):\mathbb{R}^d\to\mathbb{R}$ with the following properties:*

1. *Every f_2 that is expressed by a one-hidden-layer quadratic network of width at most $\frac{1}{2}c_1e^{c_2d}$ has*
$$\mathbb{E}_{\boldsymbol{x}\sim\mu}[f_2-\tilde{g}_{ip}]^2\geq\epsilon_1. \tag{5}$$
2. *There exists a function f_1 that is expressed by a one-hidden-layer conventional network that can approximate $\tilde{g}_{ip}$ with C_1c_σ neurons.*

Based on our proof, the relation between ϵ_1 and c_1,c_2 is characterized by $\epsilon_1\leq\|\tilde{g}_{ip}\|_{L_2(\mu)}\sqrt{2(c_1/2-k\exp(-c_2d)}$, while $0<\epsilon_1\leq\|\tilde{g}_r\|_{L_2(\mu)}\sqrt{2(c_3/2-k\exp(-c_4d)}$ according to Theorem 1 of [7].

Theorem 3 (Theorem 1 of [7])**.** *There exist universal constants $c_\sigma,c_3,c_4,C_2,\epsilon_2>0$ such that the following holds: For every dimension d, there exists a measure μ and a radial function $\tilde{g}_r(\boldsymbol{x}):\mathbb{R}^d\to\mathbb{R}$ with the following properties:*

1. *Every f_1 that is expressed by a one-hidden-layer conventional network of width at most $\frac{1}{2}c_3e^{c_4d}$ has*
$$\mathbb{E}_{\boldsymbol{x}\sim\mu}[f_1-\tilde{g}_r]^2\geq\epsilon_2. \tag{6}$$
2. *There exists a function f_2 expressed by a one-hidden-layer quadratic network that can approximate $\tilde{g}_r$ with $C_2c_\sigma d^{3.75}$ neurons.*

Proof: The detailed proof can be found in Theorem 1 of [7] and Proposition 13 of [8]. ■

The key idea of constructing $\tilde{g}_{ip}$. The purpose of construction is to enlarge the difference between $\tilde{g}_{ip}$ and f_2. First, since $\hat{f}_2(\boldsymbol{\omega})$ is radial, $\hat{\tilde{g}}_{ip}(\boldsymbol{\omega})$ is preferred to be supported over as small area as possible such that $\hat{f}_2$ and $\hat{\tilde{g}}_{ip}$ have a small overlap. Therefore, $\tilde{g}_{ip}$ is cast as an inner-product based function whose Fourier transform is only supported over a line. Second, still because $\hat{f}_2(\boldsymbol{\omega})$ is radial, we wish $\tilde{g}_{ip}$ to have unneglectable high-frequency elements because the portion of

overlap between f_2 and $\tilde{g}_{ip}$ is lower in high-frequency domain than in low-frequency domain. Specifically, given a unit vector $\boldsymbol{v}$, let $\tilde{g}_{ip}(\boldsymbol{x}) = \frac{1}{2\pi}\sqrt{\frac{4R_d}{1-\delta}}\mathrm{sinc}\Big(\frac{2R_d}{1-\delta}\boldsymbol{x}^\top\boldsymbol{v}\Big), \delta \in [0,1]$. In the following, Lemma 1 implies that $\tilde{g}_{ip}$ has certain high-frequency elements, and Lemma 2 suggests that an inner-product based function and a radial function is incompatible, since their inner-product is upper bounded exponentially.

Lemma 1. *Given a unit vector $\boldsymbol{v}$, $\tilde{g}_{ip}(\boldsymbol{x}) = \frac{1}{2\pi}\sqrt{\frac{4R_d}{1-\delta}}\mathrm{sinc}\Big(\frac{2R_d}{1-\delta}\boldsymbol{x}^\top\boldsymbol{v}\Big)$ satisfies $\int_{2R_d\mathbb{B}_d}\hat{\tilde{g}}_{ip}^2(\boldsymbol{\omega})d\boldsymbol{\omega} = 1-\delta, \delta\in[0,1]$.*

Proof: Denote the support of $\hat{\tilde{g}}_{ip}(\boldsymbol{\omega})$ as $\mathrm{Span}\{\boldsymbol{v}\}$, which is $\{\boldsymbol{\omega} = t\boldsymbol{v} \mid t\in\mathbb{R}\}$. $\tilde{g}_{ip}(\boldsymbol{x})$ is mathematically formulated as

$$\hat{\tilde{g}}_{ip}(\boldsymbol{\omega}) = \begin{cases} \sqrt{\frac{1-\delta}{4R_d}}, & \boldsymbol{\omega} = t\boldsymbol{v}, t\in[-\frac{2R_d}{1-\delta}, \frac{2R_d}{1-\delta}] \\ 0, & \boldsymbol{\omega} = t\boldsymbol{v}, t\notin[-\frac{2R_d}{1-\delta}, \frac{2R_d}{1-\delta}]. \end{cases} \tag{7}$$

Because $\hat{\tilde{g}}_{ip}$ is a constant over $[-\frac{2R_d}{1-\delta}, \frac{2R_d}{1-\delta}]$, we have $\int_{2R_d\mathbb{B}_d}\hat{\tilde{g}}_{ip}^2(\boldsymbol{\omega})d\boldsymbol{\omega} = \int_{2R_d\mathbb{B}_d}\frac{1-\delta}{4R_d}d\boldsymbol{\omega} = \frac{1-\delta}{4R_d}\int_{[-2R_d,2R_d]}dt = 1-\delta$, which concludes the proof. ∎

Lemma 2. *Let $g, f:\mathbb{R}^d\to\mathbb{R}$ be two functions of unit norm. Moreover, $\int_{2R_d\mathbb{B}_d} g^2(\boldsymbol{x})d\boldsymbol{x}\le 1-\delta$, $\delta\in[0,1]$, g is supported over $\mathrm{Span}\{\boldsymbol{v}\}+R_d\mathbb{B}_d$, and f is a combination of k radial functions, denoted as $f=\sum_{i=1}^k m_i(\|\boldsymbol{x}\|)$, where $m_i(\|\boldsymbol{x}\|)$ is a continuous radial function. Then,*

$$\langle f, g\rangle_{L_2} \le 1-\frac{\delta}{2}+k\exp(-cd/2),$$

where $c>0$ is a constant dependent on f.

Proof: Let $T = \mathrm{Span}\{\boldsymbol{v}\}+R_d\mathbb{B}_d$. For any radius $r>0$, define $h(r) = \frac{Area\left(r\mathbb{S}^{d-1}\cap T\right)}{Area(r\mathbb{S}^{d-1})}$, where $\mathbb{S}^{d-1}$ is the unit Euclidean sphere in $\mathbb{R}^d$, and $Area(\cdot)$ is to compute the area. The geometric meaning of $h(r)$ is the ratio of the area of intersection of T and $r\mathbb{S}^{d-1}$ vs the area of $r\mathbb{S}^{d-1}$. Because T is a hyper-cylinder, following the illustration in page 7 of [8], $h(r)$ is exponentially small, *i.e.,* $\exists\, c>0$, such that

$$h(2R_d)\le\exp(-cd). \tag{8}$$

As r increases, the area of the intersection decreases but the area of the sphere increases. Therefore, $h(r)$ is monotonically decreasing.

Our goal is to show $\langle f,g\rangle_{L_2}$ can be upper bounded. We decompose the inner product $\langle f,g\rangle_{L_2}$ into the integrals in a hyperball $2R_d\mathbb{B}_d$ and the region out of the hyperball $(2R_d\mathbb{B}_d)^C$. Mathematically,

$$\langle f,g\rangle_{L_2} = \int_{2R_d\mathbb{B}_d} f(\boldsymbol{x})g(\boldsymbol{x})d\boldsymbol{x} + \int_{(2R_d\mathbb{B}_d)^C} f(\boldsymbol{x})g(\boldsymbol{x})d\boldsymbol{x}. \tag{9}$$

Now, we calculate the above two integrals, respectively. For the first integral, we bound it as follows:

$$\begin{aligned}\int_{2R_d\mathbb{B}_d} f(\boldsymbol{x})\,g(\boldsymbol{x})\,d\boldsymbol{x} &\overset{(1)}{\le} \left\|g\cdot\mathbf{1}_{\{2R_d\mathbb{B}_d\}}\right\|_{L_2}\|f\|_{L_2} \\ &\overset{(2)}{\le} \sqrt{1-\delta},\end{aligned} \tag{10}$$

(1) follows from the Cauchy-Schwartz inequality; (2) follows from the fact that $f(\cdot)$ has unit L_2 norm and $\int_{2R_d\mathbb{B}_d} g^2(\boldsymbol{x})d\boldsymbol{x}\le 1-\delta$.

For the second integral, we have $\int_{(2R_d\mathbb{B}_d)^C} f(\boldsymbol{x})\,g(\boldsymbol{x})\,d\boldsymbol{x} = \int_{(2R_d\mathbb{B}_d)^C\bigcap T} f(\boldsymbol{x})\,g(\boldsymbol{x})\,d\boldsymbol{x}$ because $g(\boldsymbol{x})=0$ when $\boldsymbol{x}\notin T$. Then, we have

$$\begin{aligned}&\int_{(2R_d\mathbb{B}_d)^C\bigcap T} f(\boldsymbol{x})\,g(\boldsymbol{x})\,d\boldsymbol{x} \\ =&\sum_{i=1}^k\int_{(2R_d\mathbb{B}_d)^C\bigcap T} m_i(\|\boldsymbol{x}\|)\,g(\boldsymbol{x})\,d\boldsymbol{x} \\ \le&\sum_{i=1}^k\left(\sqrt{\int_{(2R_d\mathbb{B}_d)^C\bigcap T} m_i^2(\|\boldsymbol{x}\|)\,d\boldsymbol{x}}\sqrt{\int_{(2R_d\mathbb{B}_d)^C\bigcap T} g^2(\boldsymbol{x})\,d\boldsymbol{x}}\right) \\ \overset{(1)}{\le}&\sum_{i=1}^k\sqrt{\int_{r\ge 2R_d}\int_{(r\mathbb{S}^{d-1})\bigcap T} m_i^2(r)\,d\mathbb{S}dr} \\ =&\sum_{i=1}^k\sqrt{\int_{r\ge 2R_d}\left(\int_{r\mathbb{S}^{d-1}} m_i^2(r)\,d\mathbb{S}\right)\cdot\Big[\frac{\int_{(r\mathbb{S}^{d-1})\cap T} m_i^2(r)\,d\mathbb{S}}{\int_{r\mathbb{S}^{d-1}} m_i^2(r)\,d\mathbb{S}}\Big]dr} \\ \overset{(2)}{=}&\sum_{i=1}^k\sqrt{\int_{r\ge 2R_d}\int_{r\mathbb{S}^{d-1}} m_i^2(r)\,d\mathbb{S}\cdot h(r)\,dr} \\ \overset{(3)}{\le}&\sum_{i=1}^k\sqrt{h(2R_d)\int_{r\ge 2R_d}\int_{r\mathbb{S}^{d-1}} m_i^2(r)\,d\mathbb{S}dr} \\ \overset{(4)}{=}&\sum_{i=1}^k\sqrt{h(2R_d)}\overset{(5)}{\le} k\exp(-cd/2).\end{aligned} \tag{11}$$

In the above, (1) follows from the fact that $g(\cdot)$ has a unit L_2 norm; (2) holds because $m_i(r)$ is radial; (3) follows from $h(r)$ is monotonically decreasing; (4) follows from $f(\cdot)$ has a unit norm; (5) follows from Eq. (8).

Combining Eqs. (9), (10) and (11), we have $\langle f,g\rangle_{L_2}\le\sqrt{1-\delta}+k\exp(-cd/2)\le 1-\frac{\delta}{2}+k\exp(-cd/2)$. ∎

*Proof of **Theorem** 2.* The proof of Theorem 2 consists of two parts: (i) the inapproximability of a quadratic network to $\tilde{g}_{ip}$; (ii) the approximability of a conventional network to $\tilde{g}_{ip}$.

(i) Define $l = \frac{\widetilde{\tilde{g}_{ip}\varphi}}{\|\tilde{g}_{ip}\varphi\|_{L_2}}$. Because for $\tilde{g}_{ip}$, we have $\int_{2R_d\mathbb{B}_d}\tilde{g}_{ip}^2(\boldsymbol{x})d\boldsymbol{x} = 1-\delta$, there must exist a universal constant $c_1\in[0,1]$ such that $\int_{2R_d\mathbb{B}_d} l^2(\boldsymbol{x})d\boldsymbol{x}\le 1-c_1$. Define the function $q = \frac{\widehat{f_2\varphi}}{\|f_2\varphi\|_{L_2}}$, where $f_2 = \sum_i^k a_i\sigma(\omega_i\boldsymbol{x}^\top\boldsymbol{x}+b_i)$ is a radial function expressed by a quadratic network. Thus, according to Lemma 2, the functions $l(\cdot)$, $q(\cdot)$ satisfy

$$\langle l(\cdot), q(\cdot)\rangle_{L_2}\le 1-\frac{c_1}{2}+k\exp(-c_2d/2), \tag{12}$$

with $c_2>0$ being a universal constant.

For every scalars $\beta_1,\beta_2>0$, we have

$$\|\beta_1 l(\cdot)-\beta_2 q(\cdot)\|_{L_2}\geqslant\frac{\beta_2}{2}\|l(\cdot)-q(\cdot)\|_{L_2}. \tag{13}$$

The reason why it holds true is as follows:

Without loss of generality, we assume that $\beta_2=1$. For two unit vectors u, v in one Hilbert space, it has

$$\begin{aligned}&\min_\beta\|\beta v-u\|^2 = \min_\beta\left(\beta^2\|v\|^2-2\beta\langle v,u\rangle+\|u\|^2\right) \\ =&\min_\beta\left(\beta^2-2\beta\langle v,u\rangle+1\right) = 1-\langle v,u\rangle^2 = \frac{1}{2}\|v-u\|^2.\end{aligned}$$

Next, combining Eqs. (12) and (13), and utilizing that q, l have unit L_2 norm, we have

$$\begin{aligned}
&\|f_2-\tilde{g}_{ip}\|_{L_2(\mu)}=\|f_2\varphi-\tilde{g}_{ip}\varphi\|=\|\widehat{f_2\varphi}-\widehat{\tilde{g}_{ip}\varphi}\| \\
=&\|(\|f_2\varphi\|)\,q(\cdot)-(\|\tilde{g}_{ip}\varphi\|_{L_2})\,l(\cdot)\| \\
\geq&\frac{1}{2}\|\tilde{g}_{ip}\varphi\|\|q(\cdot)-l(\cdot)\|=\frac{1}{2}\|\tilde{g}_{ip}\|_{L_2(\mu)}\|q(\cdot)-l(\cdot)\|_{L_2} \\
\geq&\frac{1}{2}\|\tilde{g}_{ip}\|_{L_2(\mu)}\sqrt{2\left(1-\langle q,l\rangle_{L_2}\right)} \\
\geq&\|\tilde{g}_{ip}\|_{L_2(\mu)}\sqrt{2\max\left(c_1/2-k\exp\left(-c_2 d\right),0\right)},
\end{aligned} \tag{14}$$

where the first inequality is due to Eq. (13). Therefore, as long as $k \leq \frac{c_1}{2}e^{c_2 d}$, $\|f_2-\tilde{g}_{ip}\|_{L_2(\mu)}$ is larger than a positive constant ϵ_1.

(ii) In contrast, f_1 can approximate $\tilde{g}_{ip}(\boldsymbol{x}) = \frac{1}{2\pi}\sqrt{\frac{4R_d}{1-\delta}}\mathrm{sinc}\left(\frac{2R_d}{1-\delta}\boldsymbol{x}^\top\boldsymbol{v}\right)$ with a polynomial number of neurons. From the proof of Theorem 1 in [67], $\forall H:\mathbb{R}\to\mathbb{R}$ which is constant outside a bounded interval $[r_1, r_2]$, there exist scalars $a, \{\alpha_i, \beta_i, \gamma_i\}_{i=1}^k$, where $k \leq c_\sigma\frac{(r_2-r_1)L}{\epsilon}$ such that the function $h(t)=a+\sum_{i=1}^k\alpha_i\cdot\sigma(\beta_i t-\gamma_i)$ satisfies that $\sup_{t\in\mathbb{R}}|H(t)-h(t)|\leq\epsilon$.

According to this theorem, we define an auxiliary function $\tilde{g}_{ip}^{(t)}(\boldsymbol{x})$ by truncating $\tilde{g}_{ip}(\boldsymbol{x})$ when $\boldsymbol{x}^\top\boldsymbol{v}\notin[-\frac{1-\delta}{R_d\epsilon},\frac{1-\delta}{R_d\epsilon}]$ to be zero, then $|\tilde{g}_{ip}^{(t)}(\boldsymbol{x})-\tilde{g}_{ip}(\boldsymbol{x})|\leq|\tilde{g}_{ip}(\boldsymbol{x})|\leq\frac{1}{2\pi}\sqrt{\frac{4R_d}{1-\delta}}/(\frac{2R_d}{1-\delta}\cdot\boldsymbol{x}^\top\boldsymbol{v})\leq\epsilon/2$ when $\boldsymbol{x}^\top\boldsymbol{v}\notin[-\frac{\sqrt{1-\delta}}{\pi\sqrt{R_d\epsilon}},\frac{\sqrt{1-\delta}}{\pi\sqrt{R_d\epsilon}}]$. Applying the aforementioned theorem to $\tilde{g}_{ip}^{(t)}(\boldsymbol{x})$, we have $f_1(\boldsymbol{x})=a+\sum_{i=1}^k\alpha_i\cdot\sigma(\beta_i\boldsymbol{x}^\top\boldsymbol{v}-\gamma_i)$ that can approximate $\tilde{g}_{ip}^{(t)}$ in terms of $\sup_{\boldsymbol{x}\in\mathbb{R}^d}|\tilde{g}_{ip}^{(t)}(\boldsymbol{x})-f_1(\boldsymbol{x})|\leq\epsilon/2$. Here, $r_1=-\frac{\sqrt{1-\delta}}{\pi\sqrt{R_d\epsilon}}$ and $r_2=\frac{\sqrt{1-\delta}}{\pi\sqrt{R_d\epsilon}}$, the number of neurons needed is no more than $c_\sigma\frac{2\sqrt{1-\delta}L}{\pi\sqrt{R_d\epsilon}}\leq c_\sigma\frac{2\times\sqrt{5.264}\sqrt{1-\delta}L}{\pi\epsilon}=C_1c_\sigma$. Because the geometric meaning of $1/R_d$ is the volume of d-hyperball, which is upper bounded by $(1/R_d)_{d=5}\approx 5.264$[1], the number of needed neurons is irrelevant to d. Furthermore, f_1 fulfills

$$\begin{aligned}
&\sup_{\boldsymbol{x}\in\mathbb{R}^d}|\tilde{g}_{ip}(\boldsymbol{x})-f_1(\boldsymbol{x})| \\
\leq&\sup_{\boldsymbol{x}\in\mathbb{R}^d}|\tilde{g}_{ip}(\boldsymbol{x})-\tilde{g}_{ip}^{(t)}(\boldsymbol{x})|+\sup_{\boldsymbol{x}\in\mathbb{R}^d}|\tilde{g}_{ip}^{(t)}(\boldsymbol{x})-f_1(\boldsymbol{x})|\leq\epsilon.
\end{aligned} \tag{15}$$

□

Proof of ***Theorem 1****.* Theorems 3 and 2 suggest that a conventional network f_1 can express $\tilde{g}_{ip}$ with a polynomial number of neurons, and a quadratic network f_2 can express $\tilde{g}_r$ with a polynomial number of neurons. Let a heterogeneous network be $f_3=f_1+f_2$, f_3 can straightforwardly express $\tilde{g}_{ip}+\tilde{g}_r$ with a polynomial number of neurons.

Next, we need to show how to deduce from $\mathbb{E}_{\boldsymbol{x}\sim\mu}[f_2-\tilde{g}_{ip}]^2\geq\epsilon_1$ to $\mathbb{E}_{\boldsymbol{x}\sim\mu}[f_2-(\tilde{g}_{ip}+\tilde{g}_r)]^2\geq\epsilon_1$. Here, we slightly abuse ϵ_1 for succinctness. Both formulas mean that there exists a gap between functions. We can rewrite $\tilde{g}_r$ as

$$\tilde{g}_r(\|\boldsymbol{x}\|)=\tilde{g}_r\circ\sqrt{\|\boldsymbol{x}\|^2}. \tag{16}$$

Thus, we can express $\tilde{g}_r$ by a quadratic neuron using a special activation function $\sigma'=\tilde{g}_r\circ\sqrt{\cdot}$. It holds that $\sigma'(x)\leq C'(1+|x|^{\alpha'})$ for certain constants C' and α', since both $\tilde{g}_r$ and s are bounded. As a result, regarding $f_2-\tilde{g}_r$ as a quadratic network, we can get $\mathbb{E}_{\boldsymbol{x}\sim\mu}[f_2-(\tilde{g}_{ip}+\tilde{g}_r)]^2=\mathbb{E}_{\boldsymbol{x}\sim\mu}[(f_2-\tilde{g}_r)-\tilde{g}_{ip}]^2\geq\epsilon_1$.

Similarly, $\tilde{g}_{ip}$ can be re-expressed by a conventional neuron with a bounded activation. We can also get $\mathbb{E}_{\boldsymbol{x}\sim\mu}[f_1-(\tilde{g}_{ip}+\tilde{g}_r)]^2\geq\epsilon_2$ from $\mathbb{E}_{\boldsymbol{x}\sim\mu}[f_1-\tilde{g}_r]^2\geq\epsilon_2$.

□.

Remark 2. Will **Theorem 1** change if we use more layers? Indeed, our theorem and the associated proof cannot be extended into deeper networks. As Table III shows, the key ingredient of our main theorem is constructing $\tilde{g}_{ip}+\tilde{g}_r$, wherein $\tilde{g}_{ip}$ can be approximated by a one-hidden-layer conventional network with a polynomial number of neurons but a one-hidden-layer quadratic network needs an exponential number of neurons, and $\tilde{g}_r$ can be approximated by a one-hidden-layer quadratic network with a polynomial number of neurons but a one-hidden-layer conventional network needs an exponential number of neurons. The main reason accounting for such exponential differences is that it is hard for a one-hidden-layer conventional network to represent a radial function and a one-hidden-layer quadratic network to represent an inner-product-based function. However, if one allows a network to have more hidden layers, a conventional network can use a polynomial number of neurons to represent the function $g(x)=x^2$ first, and then use x^2 to construct a radial function with a polynomial number of neurons. Thus, the total number of neurons is still polynomial. In this light, although heterogeneous networks still enjoy the advantage of efficiency in representing $\tilde{g}_{ip}+\tilde{g}_r$, the difference is not exponential. A similar analysis also holds for $\tilde{g}_{ip}$ and the quadratic. We leave the extension to deeper networks as our future work.

Despite assuming a simple structure, our main theorem holds significant implications for unsupervised anomaly detection. As per the manifold hypothesis, anomalies often exhibit evident abnormal features in a low-dimensional space but become hidden or unnoticeable in a high-dimensional space [68]. Autoencoders are employed to extract the latent features of the data, thereby amplifying the differences between normal instances and anomalies [20]. When dealing with high-dimensional and complicated data, particularly in the context of unsupervised learning, our theorem assures that heterogeneous networks can approximate more flexible, more expressive, and nonlinear decision functions, thereby demonstrating superior representational ability compared to conventional networks. Furthermore, in our primary experiments, encoders and decoders for anomaly detection are designed to consist of a single hidden layer. Such heterogeneous networks indeed achieve competitive performance, as suggested by our theory, thereby showcasing high efficiency in the design of anomaly detection autoencoders.

IV. Heterogeneous Autoencoder

Given the input $\boldsymbol{x}\in\mathbb{R}^n$, let $E(\cdot)$ be the encoder and $D(\cdot)$ be the decoder, an autoencoder attempts to learn a mapping from the input to itself by optimizing the following loss function: $\min_{\boldsymbol{W}}\mathcal{J}(\boldsymbol{W};\boldsymbol{x})=\|D(E(\boldsymbol{x}))-\boldsymbol{x}\|^2$, where W is a

[1]https://en.wikipedia.org/wiki/Volume_of_an_n-ball

collection of all network parameters for simplicity. $\boldsymbol{z} = E(\boldsymbol{x})$ is the learned embedding in the low-dimensional space. The objective of the autoencoder is to learn $\boldsymbol{z}$ such that the output layer can accurately reproduce the original input. Within this structure, we refer to autoencoders that only use conventional neurons or quadratic neurons in the encoder and decoder as homogeneous autoencoders, which are referred to as AE and QAE, respectively.

Our theoretical derivation suggests combining the conventional and quadratic neurons in a network is a promising direction to explore. But it does not provide specific guidelines on which structure is the best for HAEs. Thus, we can only heuristically design the structures of the autoencoder based on a symmetry consideration: 1) utilizing equal depth and width in the encoder and decoder, which fulfills the standard practice of autoencoders; 2) arranging conventional and quadratic layers in a symmetric fashion to put them on an equal booting, since we don't have any prior knowledge which neuron type is more important. Symmetric consideration can naturally result in three heterogeneous autoencoder designs, referred to as HAE-X, HAE-Y, and HAE-I, respectively, as shown in Figure 2.

- The HAE-X conjugates a conventional and a quadratic branch in the encoder which are fused in the intermediate layer by summation. Then, this sum is transmitted to a conventional and a quadratic branch in the decoder. Next, the yields of two branches in the decoder are summed again to produce the final model output. This approach not only facilitates the heterogeneity in the model but also preserves its existing backbone.
- The HAE-Y is a simplification of the HAE-X, which respects the HAE-X in the encoder but only has a quadratic branch in the decoder. The reason why we put quadratic neurons into the decoder is that quadratic neurons are more capable of information extraction, which can help reconstruct the input more accurately. HAE-Y explores how the model's outputs are affected when an encoder and a decoder possess a similar number of neurons but different types of neurons. This serves to reduce the number of parameters and complexity of heterogeneous autoencoders.
- Other than using conventional and quadratic layers in parallel as the HAE-X and HAE-Y models do, the HAE-I interlaces conventional and quadratic layers in both an encoder and a decoder while keeping the symmetry. In the HAE-I model, we make the first layer a quadratic layer, as the quadratic layer is capable of extracting more information and ensures sufficient information to pass for subsequent layers. Since hardly we know which heterogeneous design is preferable, we examine them all in the experiments.

Training strategies. The training process of a network with quadratic neurons suffers the risk of collapse due to the high nonlinearity [6]. For example, the output function of a quadratic network with L layers will be a polynomial of 2^L degrees, which is too high to keep the network stable during training. To fix this issue, Fan *et al.* [6] proposed the so-called ReLinear algorithm, where the parameters in a quadratic neuron are initialized as $\boldsymbol{w}^g = 0, \boldsymbol{w}^b = 0, \boldsymbol{c} = 0$ and $\boldsymbol{b}^g = 1$, while $\boldsymbol{w}^r$ and b^r follow the random initialization.

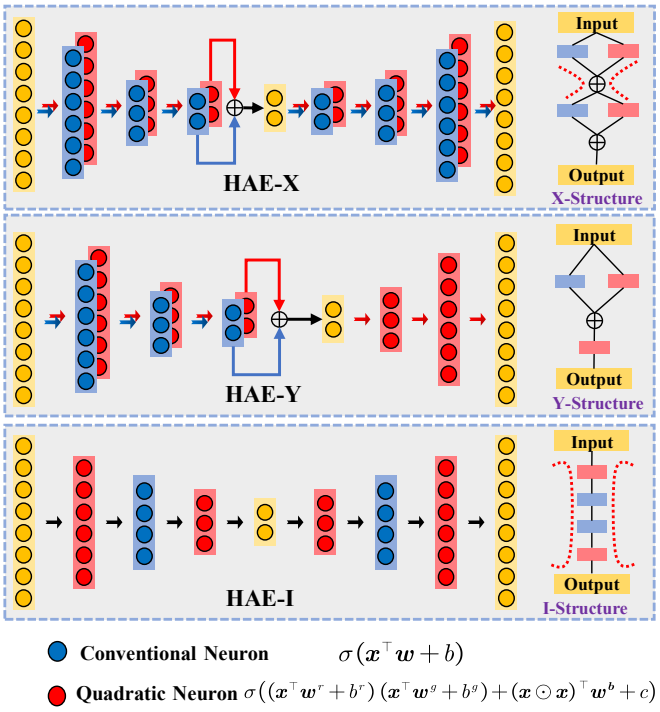


Fig. 2: Three heterogeneous autoencoder designs.

Consequently, during the initialization stage, a quadratic neuron degenerates to a conventional neuron. Then, during the training stage, different learning rates are cast for $(\boldsymbol{w}^r, b^r)$ and $(\boldsymbol{w}^g, \boldsymbol{w}^b, c, b^g)$: a normal learning rate for the former and a relatively smaller learning rate for the latter. By doing so, the nonlinearity of quadratic terms are constrained, and the learned quadratic network can avoid the training instability. The schematic illustration of the ReLinear algorithm is shown in Figure 3.

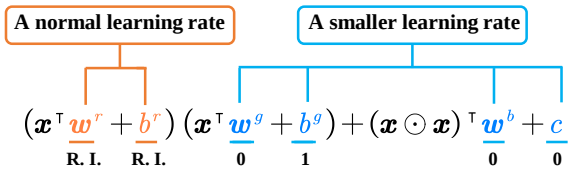


Fig. 3: ReLinear: a training strategy of quadratic networks (R. I. means random initialization).

V. Experiments

In this section, we apply the HAE to solve the anomaly detection task, by which we can also investigate if the HAE is a powerful model as suggested by theory. The anomaly detection is to distinguish between the outliers and the normal via a threshold of contamination percentage after calculating the outlier score of each sample with a model. The experiments are twofold. First, we compare HAE with a purely conventional or quadratic autoencoder. Second, we compare the HAE with either classical or SOTA methods. The results show that the HAE outperforms its competitors on 8 benchmarks. All quadratic neurons, except those used in an ablation study, are

based on the standard quadratic neurons (Eq. 1). Our code is available for readers' free download and evaluation [2].

Tabular datasets descriptions. We choose eight datasets from ODDS (Outlier Detection DataSets)[3]: multi-dimensional point datasets and seven datasets from DAMI[4] [69], which consist of a variety of tabular datasets from different fields. We eliminate duplicate datasets from DAMI repositories. Data in these datasets are labeled into two classes: normal and abnormal. For instance, the Wisconsin-Breast Cancer (WBC) dataset includes measurements for breast cancer cases, classified as either benign or malignant. The characteristics of all datasets are summarized in Table IV. In all experiments, we use 80% normal samples for training and contaminated data for testing. The test set includes normal samples that are not used in the training set and all abnormal samples.

TABLE IV: Statistics of anomaly detection benchmark datasets.

Repositories	Datasets	#Samples	#Features	Outlier Ratio
OODs	Arrhythmia	452	274	14.60%
	Glass	214	9	4.21%
	Musk	3062	166	3.16%
	Optdigits	5216	64	2.88%
	Pendigits	6870	16	2.27%
	Pima	768	8	34.90%
	Verterbral	240	6	12.50%
	Wbc	378	30	5.56%
DAMI	ALOI	50000	27	3.01%
	Ionosphere	351	7	35.89%
	KDDCUP99	60632	41	0.41%
	Shuttle	1013	9	1.28%
	Waveform	3443	21	2.90%
	WDBC	367	30	2.72%
	WPBC	198	33	23.7%

Bearing datasets descriptions. We conduct experiments on bearing fault detection, which is a high-dimensional problem. Bearings are widely used in rotating machinery, but they are prone to damage. Anomaly detection determines whether the bearing is working under normal conditions, which is the key to bearing health management. Different from tabular data, bearing fault detection is based on long vibration signals. We use two datasets of run-to-failure bearing data to validate our method: one from the public[70] and the other provided by a nuclear power plant in Hainan Province, China.

(1) XJTU-SY dataset is collected by the Institute of Design Science and Basic Component at Xi'an Jiaotong University (XJTU) and the Changxing Sumyoung Technology Co., Ltd. (SY) [70]. Two accelerometers of type PCB 352C33 (sampling frequency set to 25.6 kHz) are mounted at 90° on the housing of the tested bearings, which measure the vibration of the bearing. The accelerometers record 1.28s (32768 points) per minute until the bearing fails completely. We select the Bearing2_5 data which runs at 2250 rpm (37.5 Hz) and 11 kN load. The bearing's lifetime is 5 h 39 min and ends with an outer race fault.

(2) The seawater booster pump (SBP) data are gathered from Hainan Nuclear Power Co., Ltd. The SBP transfers cooling water to steam turbines and serves as a crucial component of the auxiliary cooling water system in nuclear power plants. Given the intricate nature of nuclear plants, mechanical systems require periodic monitoring to ensure their reliability. Figure 4 illustrates that acceleration sensors are placed at four points on both the pump and the motor when performing measurement. The vibration signal for the seawater booster pumps is measured every two months, with a sampling rate of 16.8 kHz and a recording period of 0.4 seconds, which results in 6,721 data points. The faulty and healthy signals are shown in Figure 5. Data in this study are collected from 2015 to 2023. During this period, a bearing fault at the outer race is observed.

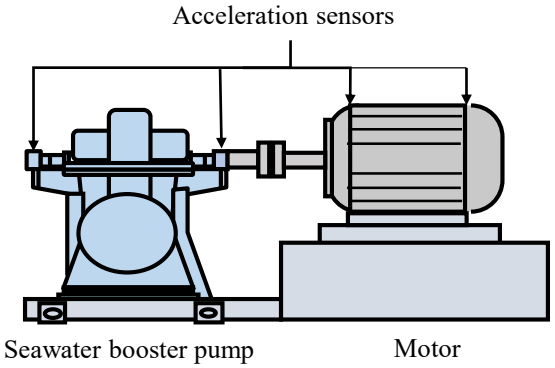


Fig. 4: The measurement structure of seawater booster pump.

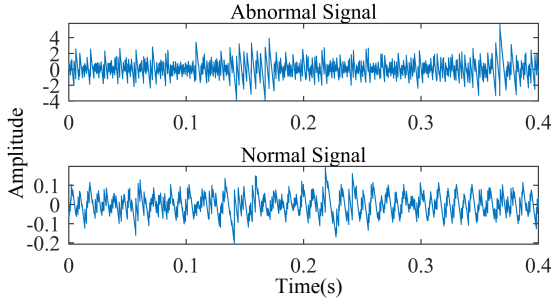


Fig. 5: Abnormal and normal signals of the seawater booster pump.

We resample all signals to 1024 points per sample and use Fast Fourier Transform to split a single side of it into 512 points. The summary of the two datasets is in Table V. Outliers in the XJTU dataset are identified based on the method described in [71]. However, it should be noted that unlike simulation tests, the availability of faulty data from real industrial scenes is very limited. As a result, only 6 samples were available for outlier detection in the SBP dataset, which makes this task pretty challenging.

Baseline Methods. We compare the proposed heterogeneous autoencoders with 7 baselines, SUDO [72], SO-GAAL [73], DeepSVDD [28], DAGMM [29], RCA [74], AE, and

[2]https://github.com/asdvfghg/Heterogeneous_Autoencoder_by_Quadratic_Neurons

[3]http://odds.cs.stonybrook.edu

[4]http://www.dbs.iĭňĄ.lmu.de/research/outlier-evaluation/DAMI

TABLE V: The summary of two bearing datasets. a×b represents the number of samples and sample dimensions.

Datasets	XJTU	SBP
#Raw Sample	334×32768	57×6721
#Abnormal Raw Sample	222×32768	1×6721
#Resample	8544×512	342×512
#Abnormal Resample	4671×512	6×512
Outlier Ratio	54.66%	1.75%

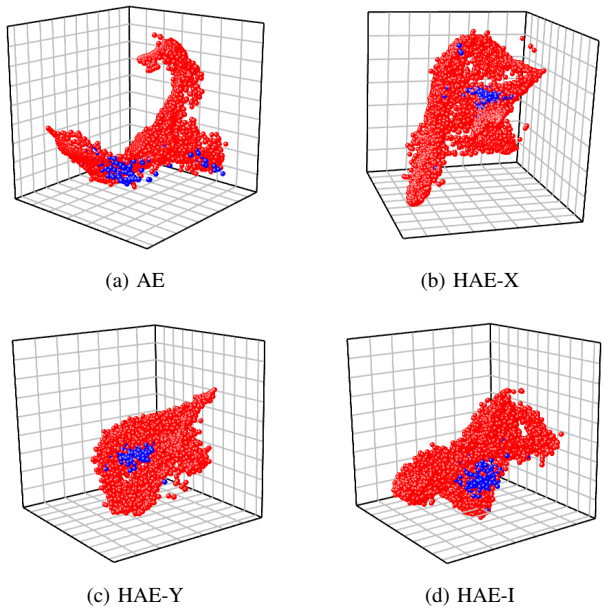
QAE (an autoencoder based on quadratic neurons). All these baselines are either classical methods with many citations or published in prestigious venues. To implement them, except for autoencoders, we utilize packages in the PyOD (Python Outlier Detection) package[5] [75] or follow official scripts[6]. We program all autoencoder models by ourselves. We use the area under the receiver operating characteristic curve (AUC) and precision (PRE) as evaluation metrics, which are widely used in unsupervised anomaly detection [27; 76; 77].

Hyperparameters. Suppose that the dimension (the number of features of the input) of the input is d, the structures of all autoencoders are $(d - \frac{d}{2} - \frac{d}{4} - \frac{d}{2} - d)$. The activation function is ReLU. The QAE and HAE are initialized with ReLinear [6], whereas the AE is initialized with a random initialization. The dropout probability is 0.5. The number of epochs is 100. We use grid search for the learning rate and batch size.

Performance comparison. Table VI and Table VII display the AUCs and PREs of all models, where we have the following observations. First, by comparing the AUCs, in the majority of datasets, HAEs are the best or second best among all benchmark methods, where HAE-X has three best performances and six second best, HAE-Y has two best and two second best. Although DeepSVDD has three best results, its AUCs are lower on other datasets. Second, HAEs demonstrated the most consistent performance on average AUC and average rank, with all three HAE models leading.At last, the same trend is observed when considering the PREs, with HAEs consistently outperforming other baselines in terms of average rank. While DeepSVDD showed competitive results on the DAMI repository, its average PRE is 1.3% lower than the best-performing HAE-X model. Note that the HAEs present varying performances on different datasets, indicating that the design of a heterogeneous autoencoder is still an open question. As a summary, our anomaly detection experiments show that the employment of diverse neurons is beneficial, supporting the earlier-developed theory for heterogeneous networks.

Table VIII presents the classification results, indicating that HAE-based methods outperform other competitors in both datasets. It should be noted that DAGMM is not included as a comparison method because its training fails. The SUOD model performs well in the SBP but achieves the limited performance in XJTU dataset, while the GAAL is the opposite. The RCA model demonstrates the best precision in the XJTU dataset and the best recall in the SBP dataset. However, it fails in some other metrics. Autoencoder-based models demonstrated better results in both datasets, implying that autoencoder-based models are relatively suitable for handling higher dimensional data. In particular, HAE models are consistently better than AE and QAE, which shows that the combination of different neuron types is a wise strategy.

Visualization. Because HAE variants show a better performance than a conventional one, we perform a learned embedding visualization to understand what these methods have learned. Here, with the optidigits dataset, we conduct t-SNE [78] to visualize the latent space learned by AE and HAEs into 3D space. As shown in Figure 6, compared to the AE result, abnormal data show distinguishable and concentration clusters by HAEs, suggesting that a heterogeneous autoencoder better HAE better expresses the divergences between the different categories of samples.



(a) AE (b) HAE-X (c) HAE-Y (d) HAE-I

Fig. 6: The t-SNE results of four autoencoders on optidigits. The red and blue points are learned latent variables from abnormal and normal data, respectively. It is observed that the outliers are more concentrated in HAEs than AE.

VI. Ablation Study

Neuron types in HAEs. Earlier, we compared the performance between HAEs and purely conventional or quadratic autoencoders. However, the proposed HAEs use different architectures from AE and QAE to better synergize the power of conventional and quadratic neurons. Consequently, it is not sure if the superior performance of HAEs is due to the synergy of different neurons or due to the architecture. To resolve this ambiguity, we replace neurons in HAEs to prototype homogeneous autoencoder, as Table IX shows. Then, we conduct experiments on 15 anomaly detection datasets to compare the difference in performance between HAEs and homogeneous autoencoders using the

[5]https://github.com/yzhao062/pyod
[6]https://github.com/illidanlab/RCA

TABLE VI: Comparison of AUCs for different anomaly detection baseline methods. Bold-faced numbers are the best and underlined numbers are the second best.

Repositories	Datasets	SUOD	SO-GAAL	DeepSVDD	DAGMM	RCA	AE	QAE	HAE-X	HAE-Y	HAE-I
OODs	Arrhythmia	0.770	0.568	0.704	0.603	0.806	0.807	0.807	**0.832**	0.816	0.817
	Glass	**0.775**	0.386	0.517	0.474	0.602	0.571	0.587	0.605	0.608	0.591
	Musk	0.568	0.438	0.502	0.903	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	Optdigits	0.469	0.367	0.494	0.290	0.622	0.652	0.599	0.772	**0.787**	0.668
	Pendigits	0.562	0.657	0.507	0.872	0.856	0.943	0.949	0.962	0.943	**0.967**
	Pima	0.624	0.292	0.485	0.531	0.711	0.657	0.652	**0.739**	0.695	0.701
	Verterbral	0.273	0.634	0.279	0.487	0.456	0.568	0.567	**0.574**	0.573	0.573
	Wbc	**0.950**	0.078	0.834	0.834	0.906	0.924	0.932	0.926	0.876	0.915
DAMI	ALOI	0.784	0.542	0.531	**1.000**	0.554	0.555	0.555	0.556	0.547	0.553
	Ionosphere	0.850	0.688	**0.979**	0.904	0.917	0.907	0.906	0.929	0.910	0.910
	KDDCUP99	0.953	0.889	0.914	**1.000**	0.989	0.989	0.986	0.989	0.989	0.989
	Shuttle	0.493	0.534	**0.892**	0.652	0.458	0.365	0.371	0.473	0.500	0.495
	Waveform	0.655	0.382	0.635	0.638	**0.704**	0.658	0.670	0.692	0.651	0.680
	WDBC	0.972	0.019	0.973	0.963	0.990	**0.997**	0.986	0.985	0.978	0.980
	WPBC	0.587	0.457	**0.703**	0.495	0.460	0.352	0.449	0.438	0.443	0.439
Avg. ↑		0.686	0.462	0.663	0.710	0.735	0.730	0.734	**0.765**	0.754	0.752
Avg. Rank ↓		6.250	8.438	7.250	6.438	4.875	5.500	5.125	**3.063**	4.313	3.750

TABLE VII: Comparison of PREs for different anomaly detection baseline methods. Bold-faced numbers are the best and underlined numbers are the second best.

Repositories	Datasets	SUOD	SO-GAAL	DeepSVDD	DAGMM	RCA	AE	QAE	HAE-X	HAE-Y	HAE-I
OODs	Arrhythmia	0.726	0.585	0.685	0.271	0.699	0.739	0.725	**0.758**	0.740	0.745
	Glass	0.593	0.516	0.573	0.497	0.596	0.572	0.576	0.582	0.589	**0.682**
	Musk	**0.794**	0.648	0.748	0.430	0.572	0.768	0.766	0.775	0.773	0.773
	Optdigits	0.448	0.447	0.473	0.513	**0.566**	0.466	0.447	0.514	0.515	0.481
	Pendigits	0.737	0.615	0.541	0.585	0.553	0.721	0.723	**0.757**	0.726	0.753
	Pima	0.638	0.382	0.511	0.567	0.541	0.591	0.636	**0.650**	0.629	0.639
	Verterbral	0.422	0.559	0.427	0.461	**0.656**	0.567	0.495	0.495	0.574	0.593
	Wbc	0.745	0.360	0.657	0.730	0.621	0.739	0.750	**0.829**	0.704	0.712
DAMI	ALOI	**0.576**	0.432	0.519	0.432	0.527	0.516	0.527	0.518	0.529	0.528
	Ionosphere	0.773	0.631	**0.915**	0.787	0.769	0.808	0.812	0.832	0.816	0.816
	KDDCUP99	0.589	0.490	0.793	0.489	0.515	0.726	0.490	0.870	**0.872**	0.871
	Shuttle	0.469	0.482	**0.835**	0.495	0.530	0.500	0.469	0.513	0.512	0.512
	Waveform	0.435	0.517	**0.655**	0.600	0.567	0.527	0.548	0.557	0.548	0.547
	WDBC	0.874	0.437	0.809	0.714	0.563	**0.993**	0.864	0.859	0.748	0.747
	WPBC	0.477	0.474	**0.674**	0.495	0.442	0.315	0.466	0.493	0.495	0.486
Avg. ↑		0.620	0.505	0.654	0.538	0.581	0.637	0.620	**0.667**	0.651	0.659
Avg. Rank↓		5.313	8.813	5.375	7.313	6.125	5.813	5.938	**3.000**	3.813	3.500

TABLE VIII: Classification results of all baseline methods over two bearing fault datasets. Bold-faced numbers are better results.

Datasets	Methods	AUC	Pre	Recall	F1
XJTU	SUOD	0.874	0.714	0.668	0.686
	SO-GAAL	0.958	0.896	0.724	0.777
	DeepSVDD	0.828	0.633	0.652	0.641
	RCA	0.946	**1.000**	0.528	0.691
	AE	0.942	0.755	0.712	0.731
	QAE	0.944	0.790	0.715	0.744
	HAE-X	0.957	0.956	**0.740**	**0.803**
	HAE-Y	0.958	0.957	0.737	0.800
	HAE-I	**0.961**	0.959	0.730	0.795
SBP	SUOD	**1.000**	0.875	0.985	0.921
	SO-GAAL	0.831	0.723	0.894	0.781
	DeepSVDD	**1.000**	0.700	0.967	0.769
	RCA	**1.000**	0.018	**1.000**	0.035
	AE	**1.000**	0.850	0.956	0.911
	QAE	**1.000**	0.873	0.963	0.904
	HAE-X	**1.000**	0.876	0.997	0.933
	HAE-Y	**1.000**	**0.883**	0.995	**0.935**
	HAE-I	**1.000**	0.881	0.990	0.932

same architecture.

TABLE IX: The structure of autoencoders with different neurons. "Q" represents quadratic neurons, "C" represents conventional neurons, "Yc" denotes the HAE-Y structure using a conventional decoder, and "Ic" denotes the HAE-I structure using the conventional layers at first.

Structures	Encoder	Decoder
HAE-X	Q+C	Q+C
QAE-X	Q+Q	Q+Q
AE-X	C+C	C+C
HAE-Y	Q+C	Q
QAE-Y	Q+Q	Q
AE-Y	C+C	C
HAE-Yc	Q+C	C
HAE-I	Q→C	C→Q
HAE-Ic	C→Q	Q→C

The results are presented in Tables X and XI. First, both AUC and precision metrics show that the performance of HAEs is superior to their homogeneous counterparts. Specifically, HAE-X achieves the highest AUC scores on 11

datasets, while HAE-Y performs best on 8 datasets and HAE-I outperforms its counterparts on 10 datasets. All quadratic neuron-based autoencoders are better than conventional ones, which underscores the powerful feature representation capabilities of quadratic neurons. At last, we observe that HAE-I with quadratic neurons in the first layer outperforms those with conventional neurons (HAE-Ic), suggesting that quadratic neurons are more adept at extracting hidden features.

The form of quadratic neurons. Quadratic neurons have different variants, such as only keeping the power terms and replacing the interaction term with a linear term. What we use in a quadratic neuron is the standard form of the quadratic function. But is it suitable for anomaly detection? Here, we investigate if the standard form of quadratic neurons fits. We have chosen several datasets for comparison, with the results for AUC and precision presented in Tables XII and XIII, respectively. We highlight that models using standard quadratic neurons display superior performance in the majority of cases. This outcome strongly suggests that the standard quadratic neuron, with its inclusion of both power and inner-product terms, possesses the best representation capabilities. Additionally, it is worth noting that models utilizing the power term generally perform better than those omitting it. This observation highlights the important role of the power term within the quadratic neuron.

TABLE XIV: Comparison of AUC for different network structures on optidigits. **Bold-faced** numbers are the best result compared in every column. For hidden neurons, V1 represents [*dim(x)*/2-*dim(x)*/4], V2 is [*dim(x)*/2-*dim(x)*/3-*dim(x)*/4], V3 is [*dim(x)*/2-*dim(x)*/3-*dim(x)*/4-*dim(x)*/4] and V4 is [*dim(x)*/2-*dim(x)*/3-*dim(x)*/3-*dim(x)*/4-*dim(x)*/4].

	AE	QAE	HAE-X	HAE-Y	HAE-I
V1	**0.652±0.017**	**0.599±0.023**	**0.680±0.054**	**0.681±0.056**	0.617±0.031
V2	0.613±0.010	0.528±0.045	0.669±0.043	0.653±0.027	**0.636±0.023**
V3	0.582±0.028	0.447±0.054	0.671±0.036	0.666±0.068	0.625±0.026
V4	0.572±0.008	0.316±0.052	0.647±0.029	0.653±0.040	0.613±0.041

VII. Analysis Experiments

Network depth. As shown in Table XIV, we investigate the performance of HAEs in response to different depths on the optidigits dataset. Note that increasing network layers will not tend to a better performance when the scale of the neural network is large enough. It may lead to over-fitting. Therefore, all autoencoders are based on V1 ([*dim(x)*/2-*dim(x)*/4]) structure in our follows experiments.

Network width. How to find the best width arrangement for autoencoders is still an open question. If the width is large, the compressing effect is compromised, which causes the encoder to fail to learn the most representative features. If the width is small, the learned features are insufficient to completely represent the original data, which makes the detection based on these features inaccurate. We conduct an experiment to evaluate the validity of different width arrangements. The results are presented in Table XV. While our recommended structure exhibits superior performance in the majority of cases, we also note that the results are quite similar. We believe that, in the context of anomaly detection where large-scale datasets are often unavailable, the number of neurons employed does not significantly impact results.

TABLE XV: Comparison of HAEs using different width arrangements in several datasets. Bold-faced numbers are better.

Dataset		Arrhythmia		Shuffle		XJTU	
Method	Structure	AUC	PRE	AUC	PRE	AUC	PRE
HAE-X	d-d/2-d/4	**0.832**	**0.758**	**0.473**	0.513	**0.957**	**0.956**
	d-d/4-d/8	0.820	0.744	0.435	0.511	0.946	0.925
	d-d-d/2	0.828	0.741	0.452	**0.520**	0.945	0.913
HAE-Y	d-d/2-d/4	0.816	0.740	**0.500**	**0.512**	**0.958**	**0.957**
	d-d/4-d/8	0.816	0.732	0.449	0.511	0.948	0.935
	d-d-d/2	**0.827**	**0.741**	0.421	0.509	0.945	0.908
HAE-I	d-d/2-d/4	**0.817**	**0.745**	**0.495**	**0.512**	**0.961**	**0.959**
	d-d/4-d/8	**0.817**	0.738	0.454	0.511	0.962	0.937
	d-d-d/2	0.814	0.737	0.445	0.508	0.958	0.947

VIII. Conclusions

In this article, we have theoretically shown that a heterogeneous network is more powerful and efficient by constructing a function that is easy to approximate for a one-hidden-layer heterogeneous network but difficult to approximate by a one-hidden-layer conventional or quadratic network. Moreover, we have proposed a novel type of heterogeneous autoencoders using quadratic and conventional neurons. Lastly, anomaly detection experiments have confirmed that a heterogeneous autoencoder delivers competitive performance relative to homogeneous autoencoders and other baselines. Future work will be designing other heterogeneous models and applying them to more real-world problems.

Acknowledgement

We would like to extend our heartfelt gratitude to Mr. Jianbin Zhu, esteemed Dean of the Vibration and Fault Diagnosis Laboratory at Hainan Nuclear Power Co., Ltd. His invaluable assistance in data collection has significantly contributed to our work. We deeply appreciate his generosity and support.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, pp. 436–444, 2015.

[2] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017, pp. 4700–4708.

[3] F.-L. Fan, D. Wang, H. Guo, Q. Zhu, P. Yan, G. Wang, and H. Yu, "On a sparse shortcut topology of artificial neural networks," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 4, pp. 595–608, 2022.

[4] M. K. Panda, B. N. Subudhi, T. Veerakumar, and V. Jakhetiya, "Modified resnet-152 network with hybrid pyramidal pooling for local change detection," *IEEE Transactions on Artificial Intelligence*, pp. 1–14, 2023.

[5] F. Fan, H. Shan, M. K. Kalra, R. Singh, G. Qian, M. Getzin, Y. Teng, J. Hahn, and G. Wang, "Quadratic autoencoder (q-ae) for low-dose ct denoising," *IEEE transactions on medical imaging*, vol. 39, no. 6, pp. 2035–2050, 2019.

[6] F.-L. Fan, M. Li, F. Wang, R. Lai, and G. Wang, "Expressivity and trainability of quadratic networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

TABLE X: Comparison of AUCs for all structures. Bold-faced numbers are the best in an identical autoencoder structure with different neurons.

Datasets	HAE-X	QAE-X	AE-X	HAE-Y	QAE-Y	AE-Y	HAE-Yc	HAE-Iq	HAE-Ic
arrhythmia	**0.832**	0.819	0.815	0.816	**0.826**	0.813	0.816	**0.817**	0.814
glass	**0.605**	0.600	0.551	**0.608**	0.585	0.554	0.559	**0.591**	0.581
musk	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
optdigits	**0.772**	0.604	0.476	**0.787**	0.656	0.475	0.487	**0.668**	0.510
pendigits	**0.962**	0.945	0.931	0.943	**0.947**	0.934	0.935	**0.967**	0.937
pima	**0.739**	0.690	0.577	**0.695**	0.606	0.582	0.581	**0.701**	0.581
verterbral	**0.574**	0.481	0.571	**0.573**	0.561	0.570	0.571	**0.573**	0.572
wbc	**0.926**	0.909	0.857	**0.876**	0.868	0.857	0.856	**0.915**	0.855
ALOI	**0.556**	0.553	0.546	**0.547**	0.554	0.546	**0.547**	**0.553**	0.547
Ionosphere	**0.929**	0.920	0.906	0.910	**0.919**	0.905	0.907	**0.910**	0.908
KDDCUP99	0.989	**0.994**	0.989	0.989	0.989	0.989	0.989	0.989	0.989
Shuttle	0.473	0.385	**0.488**	**0.500**	0.453	0.491	0.491	0.495	0.495
Waveform	**0.692**	0.681	0.643	0.651	**0.688**	0.655	0.647	**0.680**	0.657
WDBC	**0.985**	0.981	0.978	0.978	**0.980**	0.979	**0.980**	0.980	**0.981**
WPBC	0.438	0.440	**0.445**	**0.443**	0.439	0.442	0.442	0.439	**0.442**
Avg.	**0.765**	0.733	0.718	**0.754**	0.738	0.719	0.721	**0.752**	0.725

TABLE XI: Comparison of PREs for all structures. Bold-faced numbers are the best in an identical autoencoder structure with different neurons.

Datasets	HAE-X	QAE-X	AE-X	HAE-Y	QAE-Y	AE-Y	HAE_Yc	HAE_Iq	HAE_Ic
arrhythmia	**0.758**	0.742	0.729	**0.740**	0.735	0.724	0.736	**0.745**	0.734
glass	**0.582**	0.576	0.556	0.589	**0.636**	0.574	0.574	**0.682**	0.572
musk	0.775	**0.776**	0.770	0.773	**0.917**	0.771	0.773	**0.773**	0.770
optdigits	**0.514**	0.443	0.431	**0.515**	0.433	0.432	0.431	**0.481**	0.431
pendigits	**0.757**	0.717	0.712	0.726	**0.818**	0.711	0.715	**0.753**	0.715
pima	**0.650**	0.626	0.557	**0.629**	0.573	0.555	0.555	**0.639**	0.553
verterbral	0.495	0.504	**0.582**	0.574	0.572	0.586	**0.589**	**0.593**	0.589
wbc	**0.829**	0.723	0.699	**0.704**	0.695	0.700	0.703	**0.712**	0.700
ALOI	**0.518**	**0.518**	0.515	**0.529**	0.517	0.516	0.516	**0.528**	0.516
Ionosphere	**0.832**	0.827	0.812	**0.816**	0.815	0.806	0.806	**0.816**	0.807
KDDCUP99	**0.870**	0.713	0.724	**0.872**	0.727	0.729	**0.872**	**0.871**	0.726
Shuttle	0.513	0.509	**0.518**	0.512	0.511	**0.519**	0.469	0.512	**0.522**
Waveform	**0.557**	0.546	0.537	0.548	**0.558**	0.541	0.544	**0.547**	0.537
WDBC	**0.859**	0.747	0.747	0.748	0.749	0.746	**0.855**	0.747	0.747
WPBC	0.493	0.487	**0.494**	**0.495**	0.490	0.484	0.487	0.486	**0.488**
Avg.	**0.667**	0.630	0.626	**0.651**	0.650	0.626	0.642	**0.659**	0.627

TABLE XII: Comparison of AUC for different quadratic functions on some datasets. **Bold-faced** numbers are the best compared in every structure.

Methods	Quadratic Function	Arrhythmia	Glass	Optdigits	ALOI	Shuffle	Waveform	XJTU	Avg.
HAE-X	$(\boldsymbol{x}^\top \boldsymbol{w}^r + b^r)(\boldsymbol{x}^\top \boldsymbol{w}^g + b^g)$	0.817	0.591	0.627	0.552	0.471	0.682	0.942	0.669
	$\boldsymbol{x}^\top \boldsymbol{w}^r + (\boldsymbol{x} \odot \boldsymbol{x})^\top \boldsymbol{w}^b + c$	0.830	0.580	0.676	0.554	0.471	0.687	0.945	0.678
	Standard	**0.832**	**0.605**	**0.772**	**0.556**	**0.473**	**0.692**	**0.957**	**0.698**
HAE-Y	$(\boldsymbol{x}^\top \boldsymbol{w}^r + b^r)(\boldsymbol{x}^\top \boldsymbol{w}^g + b^g)$	0.820	0.578	0.659	0.553	0.453	0.689	0.942	0.671
	$\boldsymbol{x}^\top \boldsymbol{w}^r + (\boldsymbol{x} \odot \boldsymbol{x})^\top \boldsymbol{w}^b + c$	**0.829**	0.571	0.661	**0.554**	0.452	**0.695**	0.943	0.672
	Standard	0.816	**0.608**	**0.787**	0.547	**0.500**	0.651	**0.958**	**0.695**
HAE-I	$(\boldsymbol{x}^\top \boldsymbol{w}^r + b^r)(\boldsymbol{x}^\top \boldsymbol{w}^g + b^g)$	**0.818**	0.567	0.627	0.551	0.450	0.671	0.944	0.661
	$\boldsymbol{x}^\top \boldsymbol{w}^r + (\boldsymbol{x} \odot \boldsymbol{x})^\top \boldsymbol{w}^b + c$	0.815	0.568	0.628	**0.553**	0.450	0.670	**0.966**	0.664
	Standard	0.817	**0.591**	**0.668**	**0.553**	**0.495**	**0.680**	0.961	**0.681**

[7] F. Fan, J. Xiong, and G. Wang, "Universal approximation with quadratic deep networks," *Neural Networks*, vol. 124, pp. 383–392, 2020.

[8] R. Eldan and O. Shamir, "The power of depth for feedforward neural networks," in *Conference on learning theory*. PMLR, 2016, pp. 907–940.

[9] A. Ng, "Sparse autoencoder," *CS294A Lecture notes*, vol. 72, no. 2011, pp. 1–19, 2011.

[10] G. Dewangan and S. Maurya, "Fault diagnosis of machines using deep convolutional beta-variational autoencoder," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 2, pp. 287–296, 2022.

[11] Y. Li, Y. Sun, K. Horoshenkov, and S. M. Naqvi, "Domain adaptation and autoencoder-based unsupervised speech enhancement," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 1, pp. 43–52, 2022.

[12] X. Zhang, H. Zhang, and Z. Song, "Feature-aligned stacked autoencoder: A novel semisupervised deep learning model for pattern classification of industrial faults," *IEEE Transactions on Artificial Intelligence*, vol. 4, no. 4, pp. 592–601, 2023.

[13] A. Majumdar, "Blind denoising autoencoder," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 1, pp. 312–317, 2019.

[14] X. Tao, S. Yan, X. Gong, and C. Adak, "Learning multi-resolution features for unsupervised anomaly localization on industrial textured surfaces," *IEEE Transactions on Artificial*

TABLE XIII: Comparison of precision for different quadratic functions on some datasets. **Bold-faced** numbers are the best compared in every structure.

Methods	Quadratic Function	Arrhythmia	Glass	Optdigits	ALOI	Shuffle	Waveform	XJTU	Avg.
HAE-X	$(\boldsymbol{x}^\top \boldsymbol{w}^r + b^r)(\boldsymbol{x}^\top \boldsymbol{w}^g + b^g)$	0.716	**0.634**	0.457	0.516	0.517	0.548	0.747	0.591
	$\boldsymbol{x}^\top \boldsymbol{w}^r + (\boldsymbol{x} \odot \boldsymbol{x})^\top \boldsymbol{w}^b + c$	0.742	0.568	0.481	0.515	**0.523**	0.554	0.818	0.600
	Origin	**0.758**	0.582	**0.514**	**0.581**	0.513	**0.557**	**0.956**	**0.637**
HAE-Y	$(\boldsymbol{x}^\top \boldsymbol{w}^r + b^r)(\boldsymbol{x}^\top \boldsymbol{w}^g + b^g)$	**0.753**	0.572	0.460	0.516	**0.522**	0.550	0.746	0.588
	$\boldsymbol{x}^\top \boldsymbol{w}^r + (\boldsymbol{x} \odot \boldsymbol{x})^\top \boldsymbol{w}^b + c$	0.744	0.572	0.475	0.515	0.520	**0.557**	0.852	0.605
	Origin	0.740	**0.589**	**0.515**	**0.529**	0.512	0.548	**0.957**	**0.627**
HAE-I	$(\boldsymbol{x}^\top \boldsymbol{w}^r + b^r)(\boldsymbol{x}^\top \boldsymbol{w}^g + b^g)$	0.741	0.573	0.455	0.514	**0.551**	0.539	0.728	0.586
	$\boldsymbol{x}^\top \boldsymbol{w}^r + (\boldsymbol{x} \odot \boldsymbol{x})^\top \boldsymbol{w}^b + c$	0.742	0.573	0.453	0.514	0.511	0.541	0.923	0.608
	Origin	**0.745**	**0.682**	**0.481**	**0.528**	0.512	**0.547**	**0.959**	**0.636**

Intelligence, pp. 1–13, 2022.

[15] H. Liu, X. Xu, E. Li, S. Zhang, and X. Li, "Anomaly detection with representative neighbors," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–11, 2021.

[16] A. Takiddin, M. Ismail, R. Atat, K. R. Davis, and E. Serpedin, "Robust graph autoencoder-based detection of false data injection attacks against data poisoning in smart grids," *IEEE Transactions on Artificial Intelligence*, pp. 1–15, 2023.

[17] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM computing surveys (CSUR)*, vol. 54, no. 2, pp. 1–38, 2021.

[18] S. Han, X. Hu, H. Huang, M. Jiang, and Y. Zhao, "Adbench: Anomaly detection benchmark," *Advances in Neural Information Processing Systems*, vol. 35, pp. 32 142–32 159, 2022.

[19] H. Hojjati and N. Armanfard, "Dasvdd: Deep autoencoding support vector data descriptor for anomaly detection," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–12, 2023.

[20] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Müller, "A unifying review of deep and shallow anomaly detection," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 756–795, 2021.

[21] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[22] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, pp. 45–66, 2004.

[23] D. A. Reynolds *et al.*, "Gaussian mixture models." *Encyclopedia of biometrics*, vol. 741, no. 659-663, 2009.

[24] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[25] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 61, no. 3, pp. 611–622, 1999.

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

[27] Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu, and G. Chen, "Ecod: Unsupervised outlier detection using empirical cumulative distribution functions," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[28] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *ICML*, 2018, pp. 4393–4402.

[29] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International conference on learning representations*, 2018.

[30] J. Yu, X. Gao, F. Zhai, B. Li, B. Xue, S. Fu, L. Chen, and Z. Meng, "An adversarial contrastive autoencoder for robust multivariate time series anomaly detection," *Expert Systems with Applications*, vol. 245, p. 123010, 2024.

[31] Y. Yao, J. Ma, S. Feng, and Y. Ye, "Svd-ae: An asymmetric autoencoder with svd regularization for multivariate time series anomaly detection," *Neural Networks*, vol. 170, pp. 535–547, 2024.

[32] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International conference on information processing in medical imaging*. Springer, 2017, pp. 146–157.

[33] R. H. Randhawa, N. Aslam, M. Alauthman, and H. Rafiq, "Evasion generative adversarial network for low data regimes," *IEEE Transactions on Artificial Intelligence*, vol. 4, no. 5, pp. 1076–1088, 2023.

[34] E. Troullinou, G. Tsagkatakis, A. Losonczy, P. Poirazi, and P. Tsakalides, "A generative neighborhood-based deep autoencoder for robust imbalanced classification," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 1, pp. 80–91, 2024.

[35] A. O. Ojo and N. Bouguila, "A topic modeling and image classification framework: The generalized dirichlet variational autoencoder," *Pattern Recognition*, vol. 146, p. 110037, 2024.

[36] Y. Pan, F. He, X. Yan, and H. Li, "A synchronized heterogeneous autoencoder with feature-level and label-level knowledge distillation for the recommendation," *Engineering Applications of Artificial Intelligence*, vol. 106, p. 104494, 2021.

[37] T. Li, Y. Ma, J. Xu, B. Stenger, C. Liu, and Y. Hirate, "Deep heterogeneous autoencoders for collaborative filtering," in *ICDM*. IEEE, 2018, pp. 1164–1169.

[38] S. Lin, M. Zhang, X. Cheng, L. Shi, P. Gamba, and H. Wang, "Dynamic low-rank and sparse priors constrained deep autoencoders for hyperspectral anomaly detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–18, 2024.

[39] Y. Qi, Z. Yang, J. Lian, Y. Guo, W. Sun, J. Liu, R. Wang, and Y. Ma, "A new heterogeneous neural network model and its application in image enhancement," *Neurocomputing*, vol. 440, pp. 336–350, 2021.

[40] H. Sadr, M. M. Pedram, and M. Teshnehlab, "Multi-view deep network: a deep model based on learning features from heterogeneous neural networks for sentiment analysis," *IEEE access*, vol. 8, pp. 86 984–86 997, 2020.

[41] P. Pham, L. T. Nguyen, N. T. Nguyen, R. Kozma, and B. Vo, "A hierarchical fused fuzzy deep neural network with heterogeneous network embedding for recommendation," *Information Sciences*, vol. 620, pp. 105–124, 2023.

[42] C. Fu, P. Chen, Y. Shen, Y. Qin, M. Zhang, X. Lin, J. Yang, X. Zheng, K. Li, X. Sun *et al.*, "Mme: A comprehensive evaluation benchmark for multimodal large language models," *arXiv preprint arXiv:2306.13394*, 2023.

[43] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language

supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.
[44] M. Cherti, R. Beaumont, R. Wightman, M. Wortsman, G. Ilharco, C. Gordon, C. Schuhmann, L. Schmidt, and J. Jitsev, "Reproducible scaling laws for contrastive language-image learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2818–2829.
[45] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.
[46] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3836–3847.
[47] Z. Xue, G. Song, Q. Guo, B. Liu, Z. Zong, Y. Liu, and P. Luo, "Raphael: Text-to-image generation via large mixture of diffusion paths," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
[48] G. Zoumpourlis, A. Doumanoglou, N. Vretos, and P. Daras, "Non-linear convolution filters for cnn-based learning," in *ICCV*, 2017, pp. 4761–4769.
[49] G. G. Chrysos, S. Moschoglou, G. Bouritsas, Y. Panagakis, J. Deng, and S. Zafeiriou, "P-nets: Deep polynomial neural networks," in *CVPR*, 2020, pp. 7325–35.
[50] G. Chrysos, S. Moschoglou, G. Bouritsas, J. Deng, Y. Panagakis, and S. P. Zafeiriou, "Deep polynomial neural networks," *IEEE TPAMI*, 2021.
[51] G. G. Chrysos, M. Georgopoulos, J. Deng, J. Kossaifi, Y. Panagakis, and A. Anandkumar, "Augmenting deep classifiers with polynomial neural networks," in *European Conference on Computer Vision*. Springer, 2022, pp. 692–716.
[52] E. A. Rocamora, M. F. Sahin, F. Liu, G. G. Chrysos, and V. Cevher, "Sound and complete verification of polynomial networks," *arXiv preprint arXiv:2209.07235*, 2022.
[53] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh *et al.*, "Mixed precision training," *arXiv preprint arXiv:1710.03740*, 2017.
[54] Y. Jiang, F. Yang, H. Zhu, D. Zhou, and X. Zeng, "Nonlinear cnn: improving cnns with quadratic convolutions," *Neural Computing and Applications*, vol. 32, no. 12, pp. 8507–8516, 2020.
[55] P. Mantini and S. K. Shah, "Cqnn: Convolutional quadratic neural networks," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 9819–9826.
[56] M. Goyal, R. Goyal, and B. Lall, "Improved polynomial neural networks with normalised activations," in *2020 IJCNN*. IEEE, 2020, pp. 1–8.
[57] J. Bu and A. Karpatne, "Quadratic residual networks: A new class of neural networks for solving forward and inverse problems in physics involving pdes," in *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*. SIAM, 2021, pp. 675–683.
[58] Z. Xu, F. Yu, J. Xiong, and X. Chen, "Quadralib: A performant quadratic neural network library for architecture optimization and design exploration," *Proceedings of Machine Learning and Systems*, vol. 4, pp. 503–514, 2022.
[59] R. Livni, S. Shalev-Shwartz, and O. Shamir, "On the computational efficiency of training neural networks," *arXiv preprint arXiv:1410.1141*, 2014.
[60] F. Fan, W. Cong, and G. Wang, "A new type of neurons for machine learning," *International journal for numerical methods in biomedical engineering*, vol. 34, no. 2, p. e2920, 2018.
[61] K. Hu, Z. Wang, S. Mei, K. A. E. Martens, T. Yao, S. J. Lewis, and D. D. Feng, "Vision-based freezing of gait detection with anatomic directed graph representation," *IEEE journal of biomedical and health informatics*, vol. 24, no. 4, pp. 1215–1225, 2019.
[62] S. Kong and C. Fowlkes, "Low-rank bilinear pooling for fine-grained classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 365–374.
[63] C. Yu, X. Zhao, Q. Zheng, P. Zhang, and X. You, "Hierarchical bilinear pooling for fine-grained visual recognition," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 574–589.
[64] H. Zheng, J. Fu, Z.-J. Zha, and J. Luo, "Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5012–5021.
[65] R. Sun, K. Hu, K. A. E. Martens, M. Hagenbuchner, A. C. Tsoi, M. Bennamoun, S. J. G. Lewis, and Z. Wang, "Higher order polynomial transformer for fine-grained freezing of gait detection," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2023.
[66] J.-X. Liao, H.-C. Dong, Z.-Q. Sun, J. Sun, S. Zhang, and F.-L. Fan, "Attention-embedded quadratic network (qttention) for effective and interpretable bearing fault diagnosis," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–13, 2023.
[67] C. Debao, "Degree of approximation by superpositions of a sigmoidal function," *Approximation Theory and its Applications*, vol. 9, no. 3, pp. 17–28, 1993.
[68] C. Fefferman, S. Mitter, and H. Narayanan, "Testing the manifold hypothesis," *Journal of the American Mathematical Society*, vol. 29, no. 4, pp. 983–1049, 2016.
[69] G. O. Campos, A. Zimek, J. Sander, R. J. Campello, B. Micenková, E. Schubert, I. Assent, and M. E. Houle, "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study," *Data mining and knowledge discovery*, vol. 30, pp. 891–927, 2016.
[70] B. Wang, Y. Lei, N. Li, and N. Li, "A hybrid prognostics approach for estimating remaining useful life of rolling element bearings," *IEEE Transactions on Reliability*, vol. 69, no. 1, pp. 401–412, 2018.
[71] X. Huang, G. Wen, S. Dong, H. Zhou, Z. Lei, Z. Zhang, and X. Chen, "Memory residual regression autoencoder for bearing fault detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2021.
[72] Y. Zhao, X. Hu, C. Cheng, C. Wang, C. Wan, W. Wang *et al.*, "Suod: Accelerating large-scale unsupervised heterogeneous outlier detection," *Proceedings of Machine Learning and Systems*, vol. 3, 2021.
[73] Y. Liu, Z. Li, C. Zhou, Y. Jiang, J. Sun, M. Wang, and X. He, "Generative adversarial active learning for unsupervised outlier detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 8, pp. 1517–1528, 2019.
[74] B. Liu, D. Wang, K. Lin, P.-N. Tan, and J. Zhou, "Rca: A deep collaborative autoencoder approach for anomaly detection," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)*, 2021.
[75] Y. Zhao, Z. Nasrullah, and Z. Li, "Pyod: A python toolbox for scalable outlier detection," *Journal of Machine Learning Research*, vol. 20, no. 96, pp. 1–7, 2019. [Online]. Available: http://jmlr.org/papers/v20/19-011.html
[76] C. C. Aggarwal and S. Sathe, *Outlier ensembles: An introduction*. Springer, 2017.
[77] X. Wang, Y. Du, S. Lin, P. Cui, Y. Shen, and Y. Yang, "advae: A self-adversarial variational autoencoder with gaussian anomaly prior knowledge for anomaly detection," *Knowledge-Based Systems*, vol. 190, p. 105187, 2020.
[78] G. E. Hinton, "Visualizing high-dimensional data using t-sne," *Vigiliae Christianae*, vol. 9, pp. 2579–2605, 2008.