This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2022.3158422

Low-rank Characteristic Tensor Density Estimation Part II: Compression and Latent Density Estimation

Magda Amiridi, Nikos Kargas, and Nicholas D. Sidiropoulos, *Fellow, IEEE*

***Abstract*—Learning generative probabilistic models is a core problem in machine learning, which presents significant challenges due to the *curse of dimensionality*. This paper proposes a joint dimensionality reduction and non-parametric density estimation framework, using a novel estimator that can explicitly capture the underlying distribution of appropriate reduced-dimension representations of the input data. The idea is to jointly design a nonlinear dimensionality reducing auto-encoder to model the training data in terms of a parsimonious set of latent random variables, and learn a *canonical* low-rank tensor model of the joint distribution of the latent variables in the Fourier domain. The proposed latent density model is non-parametric and "universal", as opposed to the predefined prior that is *assumed* in variational auto-encoders. Joint optimization of the auto-encoder and the latent density estimator is pursued via a formulation which learns both by minimizing a combination of the negative log-likelihood in the latent domain and the auto-encoder reconstruction loss. We demonstrate that the proposed model achieves very promising results on toy, tabular, and image datasets on regression tasks, sampling, and anomaly detection.**

***Index Terms*—Statistical learning, Probability Density Function estimation, Autoencoder-based Generative Models, Dimensionality Reduction, Characteristic Function (CF), Tensors, Rank, Canonical Polyadic Decomposition (CPD).**

I. Introduction

Accurate modeling of the multivariate structure of data based on observed data samples is one of the most fundamental topics in machine learning. A model of the joint probability density function (PDF) of a data vector encodes the complete statistical properties of the data generative process and allows one to reason about data probabilistically, uncover the (possibly low-dimensional) manifold the data live on, and ultimately generate new data. PDF estimation serves as a building block in a wide variety of applications, such as image processing [1], speech modeling [2], natural language processing [3], and anomaly detection [4]. Conventional density estimation methods, such as kernel density estimation (KDE) [5] and Gaussian mixture models (GMMs) [6] are usually designed to fit target distributions directly in the data space $\mathbb{R}^N$ and fall short in high dimensions from both computational and statistical points of view due to the *Curse of Dimensionality* – convergence slows down as the number of dimensions increases as a result of data sparsity in high-dimensional spaces. Real-world data often resides in a high-dimensional and complex feature space with only a limited amount of observed data being directly available.

Recently, the use of deep neural networks has led to substantial advances in this area. For example, generative adversarial networks (GANs) [7] can be trained to sample from very high-dimensional densities, but they do not support statistical inference or explicit density evaluation. On the other hand, variational auto-encoders (VAEs) [8] provide functionality for both (approximate) inference and sampling. VAEs assume a prior as a manually specified distribution (e.g., a simple isotropic Gaussian or mixture of Gaussians) and are trained by minimizing a reconstruction error and a divergence to force the variational posterior to fit the prior of the latent variables. However, the forced global structure in the latent space through the use of a manually specified prior may differ from the complex latent nature of the true data manifold. Thus, such simplistic assumptions may potentially harm the generalization of high dimensional data from low dimensional latent spaces. For example, it is observed that VAEs tend to generate blurry images, an effect that is usually attributed to the latent density mismatch problem [9], [10]. Finally, explicit neural models such as auto-regressive models [11] and flow-based models [12], [13] are designed to perform sampling and point-wise density evaluation. Despite their success, auto-regressive models generally suffer from slow sampling time [14] and inferior quality of samples compared to VAEs; but they are particularly useful for point-wise density evaluation. On the other hand, flow-based models, such as Real-NVP [13] and Glow [1], are efficient for sampling, but have inferior performance in evaluating the log-likelihood of the input compared to the auto-regressive ones.

The goal of this paper is to introduce a class of probabilistic latent variable models for unsupervised learning which is tailored for high dimensional datasets. The proposed class of models is non-parametric, and it learns the underlying distribution of latent representations of the input data in the Fourier domain. The proposed framework consists of two main components: an auto-encoder network, through which a lower-dimensional latent representation of the input is sought; and a nonparametric density estimation module in the latent domain. The auto-encoder compresses redundancies in the data domain while preserving the essential information, and is used as a new feature representation space where we learn the data distribution. The auto-encoder and the latent density are learned jointly via an optimization criterion that combines a data reconstruction loss and a negative log-likelihood regularization term over the latent representations of the training data.

This is the second part of a two-part paper. The first part

Original manuscript submitted June 19, 2021; revised March 3, 2022; accepted March 5, 2022. This work was supported in part by NSF IIS-1704074. M. Amiridi and N.D. Sidiropoulos are with the Department of ECE, University of Virginia, Charlottesville, VA 22904. Author e-mails: (ma7bx,nikos)@virginia.edu

N. Kargas was with the Department of ECE, University of Minnesota; he is now with Amazon, Cambridge, U.K. Author e-mail: karga005@umn.edu

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2022.3158422

[15] dealt with the density estimation problem in the native ("raw") input data domain, showing that any joint density that is compactly supported and continuously differentiable can be well-approximated using a low-rank tensor model in the Fourier domain. A corollary of [15] is that a finite separable mixture model (approximately) follows from compactness of support and continuous differentiability. This interpretation enables an efficient and disciplined sampling process. By introducing a low-rank tensor model in the Fourier domain via the Canonical Polyadic Decomposition (CPD) [16], a controllable approximation of the multivariate density is identifiable. The choice of tensor rank, number of Fourier coefficients, and the dimensionality of the latent space let us control the expressivity of the learned distribution. With respect to Part I [15], the key differences in this second part are the following:

- Unlike Part I, which aimed to tackle the problem directly in the original N-dimensional space, probabilistic modeling in Part II is realized in a reduced-dimension latent space and the effects are translated back into the input space through the decoder mapping. Towards this end, a *joint* nonlinear dimensionality reduction and compressed density estimation framework is proposed in Part II. The joint approach boosts the flexibility, scalability, and statistical performance in terms of prediction (regression/detection) accuracy and sampling fidelity.
- Instead of the coupled tensor factorization approach adopted in Part I, Part II tackles joint density estimation as a *hidden tensor factorization* problem using maximum likelihood learning of the latent distribution's parameters.

A high-level overview of the proposed framework is shown in Figure 1. A sneak peak of the expected performance of the proposed method is shown in Figure 2 where we use our model to learn the joint distribution of MNIST [17] images of 0s and 8s. With a suitable combination of hyper-parameters, the proposed density estimator offers considerable flexibility without sacrificing parsimony of representation. We showcase the promising results of the proposed model on benchmark image (MNIST, FMNIST [18]) and several tabular datasets on sampling, regression, and anomaly detection tasks, and on some toy but didactic examples for illustration.

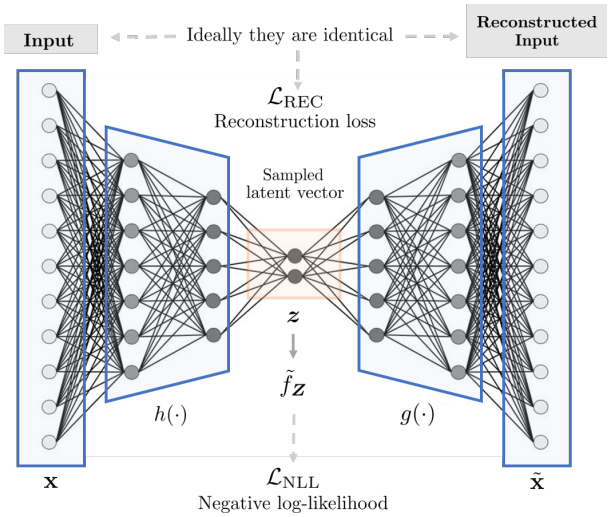


Fig. 1: Compressed Density Estimation: An auto-encoder attempts to reproduce the input in the output layer by compressing it to fewer dimensions while retaining non-redundant information. The hidden layer becomes a bottleneck, forming a lower-dimensional representation of the data, which is used to build a non-parametric density model.

II. Background

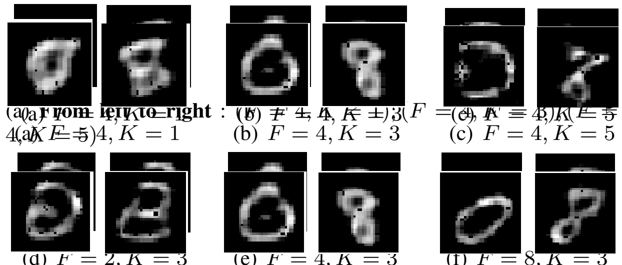
A. Related work

Classic work on density estimation includes Gaussian Mixture Models, which are fragile to model mismatch due to their parametric nature, and introduce computational and estimation challenges in the high dimensional case. Conventional non-parametric models such as the Kernel Density Estimators become computationally intractable in high dimensions, since the number of parameters grows exponentially with the number of dimensions [19].

Recently, the use of deep neural networks has led to significant advances in modeling modern complex and high-dimensional data. Auto-encoders enjoy a remarkable ability to learn data representations. Auto-encoder networks such as VAEs [8] and GANs [7] learn latent representations of very high-dimensional data such as images or videos. However, GANs only support sampling, but not inference or density estimation. VAEs assume that high-dimensional data can be modeled as lying on or near a low-dimensional, nonlinear manifold which they approximate by learning nonlinear mappings while encouraging a global structure in the latent space through the use of a specified prior distribution. However, specifying the prior distribution may prevent them from faithfully representing the true data manifold. It was shown in [20] that choosing a simplistic prior could lead to over-regularization and, as a consequence, very poor latent representation.

In this work, we avoid the variational training by jointly training a deterministic encoder–decoder pair and an expressive density estimator in the latent space, which admits simpler optimization, and, most importantly, generates better samples than VAEs. A key advantage of our approach is that we introduce a non-parametric density model into the latent space, which by virtue of uniqueness of low rank tensor decomposition comes with identification guarantees. This approach can yield a more accurate model of the data manifold, as we will see. A conceptually similar approach was proposed in [4] and applied for unsupervised anomaly detection, the key difference being that the density of low-dimensional representations was modelled using a GMM, which is far more restrictive and does not come with identification guarantees.

Other classes of generative models include the Real-valued Neural Autoregressive Distribution Estimator (RNADE) [21] and its discrete version MADE [22], which is among the best performing neural density evaluation methods and has shown great potential in scaling to high-dimensional distribution evaluation problems. These so-called autoregressive models decompose the joint density as a product of one-dimensional conditionals of increasing conditioning order, and model each conditional density with a parametric model. Normalizing



(a) $F=4, K=1$ (b) $F=4, K=3$ (c) $F=4, K=5$

(d) $F=2, K=3$ (e) $F=4, K=3$ (f) $F=8, K=3$

Fig. 2: Sneak peek: Demonstration of generated MNIST samples trained on images of 0s and 8s using the proposed CDE model. We train the model on different values of F and K to show that only a few parameters are needed to come close to the ground-truth. Increasing K generates sharper digits, while increasing F better differentiates the samples of the digits.

type of NF model where the transformation layer is built as an autoregressive neural network. Finally, Gaussianization flows (GF) [25] build upon rotation-based iterative Gaussianization. These methods do not construct an explicit joint PDF model, but rather serve for point-wise density evaluation. That is, for any given input vector (realization), they output an estimate of the density evaluated at that particular input vector.

B. Notation

In this paper, we use $\mathbf{x}$, $\mathbf{X}$, $\underline{\mathbf{X}}$ for vectors, matrices and tensors respectively. We use the notation $\mathbf{x}(k)$, $\mathbf{X}(:,k)$, $\underline{\mathbf{X}}(:,:,k)$ to refer to a particular element of a vector, a column of a matrix and a slab of a tensor. Symbols $\|\mathbf{x}\|_2$ and $\|\mathbf{X}\|_F$ correspond to L_2 norm and Frobenius norm. Symbols $\circ$, $\circledast$, $\odot$ denote the outer, Hadamard and Khatri-Rao product respectively. The set of integers $\{1,\ldots,N\}$ is denoted as $[N]$. We use the notation FC(a,b,c) to describe a fully connected layer with a input neurons and b output neurons activated by function c.

C. Canonical Polyadic Decomposition

In this section, we briefly introduce basic concepts related to tensor decomposition. A D-way tensor $\underline{\mathbf{\Phi}} \in \mathbb{C}^{K_1\times K_2\times\cdots\times K_D}$ is a multidimensional array whose elements are indexed by D indices. Any tensor can be decomposed as a sum of F rank-1 tensors, i.e.

$$\underline{\mathbf{\Phi}} = [\![\boldsymbol{\lambda}, \mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_D]\!] = \sum_{f=1}^{F} \boldsymbol{\lambda}(f)\mathbf{a}_f^1 \circ \mathbf{a}_f^2 \circ \cdots \circ \mathbf{a}_f^D, \quad (1)$$

where $\mathbf{A}_d = [\mathbf{a}_1^d, \ldots, \mathbf{a}_F^d] \in \mathbb{C}^{K_d\times F}$ and constraining the columns $\mathbf{A}_d(:,f)$ to have unit norm, the real scalar $\boldsymbol{\lambda}(f)$ absorbs the f-th rank-one tensor's scaling. A particular element of the tensor is given by

$$\underline{\mathbf{\Phi}}(k_1, k_2, \ldots, k_D) = \sum_{f=1}^{F} \boldsymbol{\lambda}(f) \prod_{d=1}^{D} \mathbf{A}_d(k_d, f). \quad (2)$$

When F is minimal, it is called the rank of $\underline{\mathbf{\Phi}}$, and the decomposition is called Canonical Polyadic Decomposition (CPD) [26], [16]. CPD is a powerful model that can parsimoniously represent the high-order interactions among multi-way data exactly or approximately leading to significant reduction in the number of parameters. A key property of the CPD is that the rank-1 components are unique under mild conditions. For learning latent variable statistical models, the uniqueness of tensor decomposition often translates to identifiability, that is, the existence of a unique set of parameters that can be consistent with what we have observed. See [27] for a tutorial overview and detailed identifiability results.

III. Compressed-domain Density Estimation

We consider the problem of general-purpose modeling of a high-dimensional continuous joint distribution $f_{\mathbf{X}}$ of an N-dimensional random vector $\mathbf{X}$ when N is large. Given a dataset $\mathcal{D}$ of M i.i.d. realizations in the N-dimensional observable space $\mathcal{D} = \{\mathbf{x}_m\}_{m=1}^{M}$, we typically wish to perform maximum likelihood learning of its parameters, i.e., to minimize the Negative Log-Likelihood (NLL)

$$\mathcal{L}_{\text{NLL}} = -\frac{1}{M}\sum_{m=1}^{M} \log\big(f_{\mathbf{X}}(\mathbf{x}_m)\big). \quad (3)$$

In general, $\mathcal{L}_{\text{NLL}}$ is difficult to compute or differentiate directly, since the density $f_{\mathbf{X}}$ can be analytically and computationally intractable. One can address this issue and evade the curse of dimensionality by using a mapping h to encode the input data samples $\mathbf{x}_m \in \mathbb{R}^N$ into much lower dimensional representations $\mathbf{z}_m \in \mathbb{R}^D$, with $D \ll N$, in the latent space.

In this work, we propose a joint dimensionality reduction (DR) and density estimation framework where the DR part is carried out through learning an auto-encoder:

$$\text{Auto-encoder:} \quad \mathbf{x} \overset{h}{\mapsto} \mathbf{z} \overset{g}{\mapsto} \tilde{\mathbf{x}}.$$

Here, h and g denote the encoder and the decoder, respectively, and $\tilde{\mathbf{x}}$ is the reconstruction of $\mathbf{x}$. The mapping $h: \mathbb{R}^N \to \mathbb{R}^D$ can be viewed as nonlinear dimensionality reduction, and the low-dimensional $\mathbf{z} = h(\mathbf{x})$ as the bottleneck representation of the observed vector $\mathbf{x}$. We approximate the latent distribution $f_{\mathbf{Z}}$ using the non-parametric density estimation framework in [15]. The density estimation framework in [15] relies on the decomposition of a D-way tensor of Fourier series coefficients through a CPD, showing that this model is quite general, in that it can approximate any multivariate compactly supported density as long as its Fourier coefficients decay sufficiently fast, and under certain conditions it can identify the true latent model.

The tensor rank, the number of Fourier coefficients, and the dimensionality of the bottleneck representation are used to control the expressivity of the model. Here we propose to *jointly learn* the auto-encoder and the parameters of the density model. The combination of an auto-encoder and density estimation takes advantage of their synergistic strengths. Auto-encoders can compress input data to fewer dimensions while retaining non-redundant information, while density estimation works best in lower-dimensional spaces. The proposed framework can be used for missing data imputation and as a generative model.

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2022.3158422

of a data tensor. Rank determination (subject to a user-specified upper bound on rank) is automatic in these methods, but it falls off statistical modeling assumptions that may or may not be appropriate in our context. Note that we model the complex tensor of multivariate Fourier series coefficients of the joint PDF in some latent space. We know that for differentiable PDFs the high-frequency coefficients will typically be much smaller than low-frequency ones; but the priors used on the factors in Bayesian tensor models are i.i.d. across modes.

Interestingly, reference [28] is about fitting non-negative tensors using non-negative factors. While is not useful for PDF modeling in the Fourier domain, it can potentially be used in the multivariate PMF case considered earlier in our work [30], if one can incorporate certain sum to one constraints in the framework of [28].

III. Compressed-domain Density Estimation

We consider the problem of general-purpose modeling of a high-dimensional continuous joint distribution $f_{\boldsymbol{X}}$ of an N-dimensional random vector $\boldsymbol{X}$ when N is large. Given a dataset $\mathcal{D}$ of M i.i.d. realizations in the N-dimensional observable space $\mathcal{D} = \{\mathbf{x}_m\}_{m=1}^{M}$, we typically wish to perform maximum likelihood learning of its parameters, i.e., to minimize the Negative Log-Likelihood (NLL)

$$\mathcal{L}_{\mathrm{NLL}} = -\frac{1}{M}\sum_{m=1}^{M} \log\Big(f_{\boldsymbol{X}}(\mathbf{x}_m)\Big). \tag{3}$$

In general, $\mathcal{L}_{\mathrm{NLL}}$ is difficult to compute or differentiate directly, since the density $f_{\boldsymbol{X}}$ can be analytically and computationally intractable. One can address this issue and evade the curse of dimensionality by using a mapping h to encode the input data samples $\mathbf{x}_m \in \mathbb{R}^N$ into much lower dimensional representations $\mathbf{z}_m \in \mathbb{R}^D$, with $D \ll N$, in the latent space.

In this work, we propose a joint dimensionality reduction (DR) and density estimation framework where the DR part is carried out through learning an auto-encoder:

$$\text{Auto-encoder:}\quad \mathbf{x} \overset{h}{\mapsto} \mathbf{z} \overset{g}{\mapsto} \tilde{\mathbf{x}}.$$

Here, h and g denote the encoder and the decoder, respectively, and $\tilde{\mathbf{x}}$ is the reconstruction of $\mathbf{x}$. The mapping $h : \mathbb{R}^N \to \mathbb{R}^D$ can be viewed as nonlinear dimensionality reduction, and the low-dimensional $\mathbf{z} = h(\mathbf{x})$ as the bottleneck representation of the observed vector $\mathbf{x}$. We approximate the latent domain distribution $f_{\boldsymbol{Z}}$ using the non-parametric density estimation framework in Part I of this work [15]. The density estimation framework relies on the decomposition of a D-way tensor of leading Fourier series coefficients through CPD. Part I has shown that this model is quite general, in that it can approximate any multivariate compactly supported density as long as its Fourier coefficients decay sufficiently fast, and under certain conditions it can identify the true latent model.

The choice of tensor rank, number of Fourier coefficients, and the dimensionality of the bottleneck representation are used to control the expressivity of the model. Here we propose to *jointly learn* the auto-encoder and the parameters of the density model. The combination of an auto-encoder and density estimation takes advantage of their synergistic strengths. Auto-encoders can compress input data to fewer dimensions while retaining non-redundant information, while density estimation works best in lower-dimensional spaces. The resulting training objective provides a strong underlying signal to efficiently capture the latent space distribution of the input data during training. The proposed framework can be used for missing data imputation and as a generative model.

Missing data imputation: Assume that for a given data sample $\mathbf{x}$, we observe a subset of its values denoted as $\mathbf{x}_O$ and $\mathbf{x}_M$ is the part that we do not observe. Data imputation can be performed by clamping the observed dimensions $\mathbf{x}_O$ to their values and maximizing log-likelihood with respect to the missing dimensions $\mathbf{x}_M$

$$\max_{\mathbf{x}_M} \log\big(f_{\boldsymbol{Z}}\big(h(\mathbf{x}_O, \mathbf{x}_M)\big)\big). \tag{4}$$

Data sampling: With g given, we can draw a realization of the random vector $\boldsymbol{Z}$ in the D-dimensional latent space from $f_{\boldsymbol{Z}}$, and back transform to a sample in the original N-dimensional space by its inverse image as

$$\mathbf{z} \sim f_{\boldsymbol{Z}},\ \tilde{\mathbf{x}} = g(\mathbf{z}). \tag{5}$$

Similar approaches such as VAEs pose a stochastic condition on the latent variables to comply with a fixed prior distribution $f_{\boldsymbol{Z}}$ over a low-dimensional latent space:

$$\text{VAE:}\quad \mathbf{x} \overset{h}{\mapsto} \mathbf{z} \overset{g}{\mapsto} \tilde{\mathbf{x}},\ \mathbf{z} \sim f_{\boldsymbol{Z}}(\mathbf{z}).$$

The generative process of the VAE is carried out as

$$\mathbf{z} \sim f_{\boldsymbol{Z}}, \mathbf{x} \sim p_\theta(\mathbf{X}|\mathbf{Z} = \mathbf{z})$$

where a stochastic decoder

$$D_\theta(\mathbf{z}) = \mathbf{x} \sim p_\theta(\mathbf{x}|\mathbf{z}) = p(\mathbf{X}|g(\mathbf{z}))$$

links the latent space to the input space through the likelihood distribution p_θ. This may be limiting in case this predefined prior does not match the structure of the true data manifold, leading to a less accurate model. Our approach is fundamentally different as we avoid prior distribution matching between the variational posterior and the prior, but instead propose jointly learning a non-parametric density estimator in the latent space. Most importantly, our approach produces better samples than VAEs, as we will see. In the following sections, we give a detailed description of the two main components of our framework and the optimization procedure.

A. Compression Network

The first component of our framework seeks a non-linear mapping h to project high-dimensional input samples into a low-dimensional space. In the dimensionality reduction process, discarding some dimensions inevitably leads to information loss. We wish to preserve the available information as much as possible, and to this end we minimize the empirical approximation of the mean squared error

$$\mathrm{MSE} := \int_{\mathbb{R}^N} \|\boldsymbol{x} - g(h(\boldsymbol{x}))\|_2^2 f_{\boldsymbol{X}}(\boldsymbol{x}) d\boldsymbol{x}. \tag{6}$$

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2022.3158422

Auto-encoders learn a function by fine-tuning the parameters of a feed-forward Deep Neural Network (DNN) in such a way that the reconstruction error is minimized when back projected with another feed-forward DNN. These networks need to be specified a-priori, in terms of the number of layers and neurons. In this work, we use the rectified linear unit (ReLU) activation function [31] while the rest of the parameters such as the width of each layer and the depth of the network are adjusted according to M, N.

Although we proceed with simpler networks, other types of networks (e.g., convolutional neural networks [32], [33]) can also be used. Let $h(\cdot;\boldsymbol{\theta}_h)$ and $g(\cdot;\boldsymbol{\theta}_g)$ be DNNs and $\boldsymbol{\theta}_h$, $\boldsymbol{\theta}_g$ collect the encoder and decoder network parameters, i.e., the weights and bias terms at each hidden layer. Given a finite set of samples M, the empirical reconstruction loss can be computed as

$$\mathcal{L}_{\text{REC}} := \frac{1}{M}\sum_{m=1}^{M} ||\mathbf{x}_m - g(h(\mathbf{x}_m;\boldsymbol{\theta}_h);\boldsymbol{\theta}_g)||_2^2. \tag{7}$$

The reconstruction loss is typically minimized using Stochastic Gradient Descent (SGD).

B. Latent Density Estimation Network

The second component of our framework is a non-parametric density estimation model [15]. The key difference is that we propose joint dimensionality reduction and density modeling in the reduced-dimension latent space so that we capture the bottleneck layer distribution, whereas [15] aimed to tackle the problem directly in the original N-dimensional space. This combination is crucial for enhanced performance and scalability. Additionally, instead of coupled tensor factorization, we consider an alternative algorithmic approach by formulating density estimation as a hidden tensor factorization problem.

Let us consider the multivariate joint PDF $f_{\boldsymbol{Z}}$ of a D-dimensional random vector $\boldsymbol{Z}$ with its support contained within the hypercube $S = [0,1]^D$. Then, the joint PDF can be represented by a multivariate Fourier series

$$f_{\boldsymbol{Z}}(\mathbf{z}) = \sum_{k_1=-\infty}^{\infty} \cdots \sum_{k_D=-\infty}^{\infty} \Phi_{\boldsymbol{Z}}[\boldsymbol{k}] e^{-j2\pi \mathbf{k}^T \mathbf{z}}, \tag{8}$$

where

$$\Phi_{\boldsymbol{Z}}[\mathbf{k}] = \Phi_{\boldsymbol{Z}}(\boldsymbol{\nu})\big|_{\boldsymbol{\nu}=2\pi\mathbf{k}}, \mathbf{k} = [k_1, \ldots, k_D]^T$$

and Φ_Z is the Characteristic Function (CF). The multivariate characteristic function $\Phi_Z : \mathbb{R}^D \to \mathbb{C}$ is defined as

$$\Phi_{\boldsymbol{Z}}(\boldsymbol{\nu}) = E\left[e^{j\boldsymbol{\nu}^T \boldsymbol{Z}}\right].$$

Similar to the PDF $f_{\boldsymbol{Z}}$, its corresponding CF $\Phi_{\boldsymbol{Z}}$ contains complete information about the distribution of $\boldsymbol{Z}$, i.e., the PDF and the CF have a bijective relationship – one being the Fourier transform of the other. When the underlying PDF is sufficiently differentiable in all variables, $f_{\boldsymbol{Z}}$ can be approximated by a truncated multivariate Fourier series with cutoffs $K_1, \ldots, K_D$ i.e.,

$$\tilde{f}_{\boldsymbol{Z}}(\boldsymbol{z}) = \sum_{k_1=-K_1}^{K_1} \cdots \sum_{k_D=-K_D}^{K_D} \Phi_{\boldsymbol{Z}}[\boldsymbol{k}] e^{-j2\pi \boldsymbol{k}^T \boldsymbol{z}}. \tag{9}$$

The smoother the underlying PDF the faster the convergence rate and the smaller the approximation error.

For any $p \in \mathbb{N}$, If the partial derivatives $\frac{\partial^{\theta_1}}{\partial z_1^{\theta_1}} \cdots \frac{\partial^{\theta_D}}{\partial z_D^{\theta_D}} f_{\boldsymbol{Z}}(\mathbf{z})$ of $f(\cdot)$ exist and are absolutely integrable for all $\theta_1, \ldots, \theta_D$ with $\sum_{n=1}^{D} \theta_n \leq p$ then the rate of decay of the magnitude of the $\mathbf{k}$-th Fourier coefficient $|\Phi_{\boldsymbol{Z}}[\mathbf{k}]|$ obeys [34]

$$|\Phi_{\boldsymbol{Z}}[\mathbf{k}]| = \mathcal{O}\left(\frac{1}{1+\|\mathbf{k}\|_2^p}\right).$$

The worst-case approximation error is bounded by

$$\|f_{\boldsymbol{Z}} - \tilde{f}_{\boldsymbol{Z}}\|_\infty \leq C \sum_{d=1}^{D} \frac{\omega_d\left(\frac{\partial^{\theta_d}}{\partial z_d^{\theta_d}} f_{\boldsymbol{Z}}, \frac{1}{1+K_d}\right)}{(1+K_d)^{\theta_d}},$$

where $C = C_2\left(1 + C_1 \prod_{d=1}^{D} \log K_d\right)$, C_1, C_2 are constants independent of $f_{\boldsymbol{Z}}$ and the K_d's and

$$\begin{aligned}\omega_j(f_{\boldsymbol{Z}}, \delta) := \sup_{|z_j - z_j'| \leq \delta} & |f_{\boldsymbol{Z}}(z_1, ..., z_j, ..., z_D) \\ & - f_{\boldsymbol{Z}}(z_1, ..., z_j', ..., z_D)|\end{aligned} \tag{10}$$

measures the smoothness of $f_{\boldsymbol{Z}}$ for each component $j \in [D]$ [35], [36, Chapter 23]. Note that we can represent the truncated Fourier coefficients using a D-way tensor $\underline{\boldsymbol{\Phi}}$ where

$$\underline{\boldsymbol{\Phi}}(k_1, \ldots, k_D) = \Phi_{\boldsymbol{Z}}[\mathbf{k}]. \tag{11}$$

For simplicity we will assume that $K_1 = \cdots = K_D = K$. Orthogonal series PDF approximation using a truncated sum of basis functions (e.g., trigonometric, polynomial, wavelet) becomes computationally intractable in high dimensions, since the number of parameters (tensor elements) grows exponentially with the number of dimensions. To reduce the number of parameters, we introduce a low-rank parameterization of the coefficient tensor obtained by truncating the multidimensional Fourier series [15] which reduces the number of parameters from $O(K^D)$ to $O(DKF)$. Introducing the rank-F CPD we have

$$\underline{\boldsymbol{\Phi}}(k_1, \ldots, k_D) = \sum_{f=1}^{F} p_H(f) \prod_{d=1}^{D} \Phi_{Z_d|H=f}(k_d|f). \tag{12}$$

By linearity and separability of the multidimensional Fourier transformation, applied to rank-one components, $\tilde{f}_{\boldsymbol{Z}}(\mathbf{z})$ can be written in the following form

$$\begin{aligned}\tilde{f}_{\boldsymbol{Z}}(\mathbf{z}) &= \sum_{k_1=-K_1}^{K_1} \cdots \sum_{k_D=-K_D}^{K_D} \Phi_{\boldsymbol{Z}}[\mathbf{k}] e^{-j2\pi \mathbf{k}^T \mathbf{z}} \\ &= \sum_{f=1}^{F} \underbrace{p_H(f)}_{\boldsymbol{\lambda}(f)} \prod_{d=1}^{D} \sum_{k_d=-K}^{K} \underbrace{\Phi_{Z_d|H=f}(k_d|f)}_{\mathbf{a}_d^f(K+1+k_d)} \underbrace{e^{-j2\pi k_d z_d}}_{\mathbf{b}_d(K+1+k_d)} \\ &= \sum_{f=1}^{F} \boldsymbol{\lambda}(f) \prod_{d=1}^{D} \mathbf{A}_d(:, f)^T \mathbf{b}_d.\end{aligned}$$

The above joint PDF $f_{\boldsymbol{Z}}$ model can be interpreted as a mixture of F product distributions, i.e., there exists a 'hidden' random variable H taking values in $\{1, \ldots, F\}$ that selects

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2022.3158422

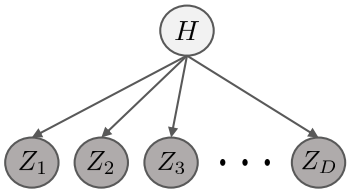


Fig. 3: Our approach yields a generative model of the latent density, from which it is very easy to sample. This is because $f_{\boldsymbol{Z}}$ can be interpreted as a mixture of F product distributions i.e., admits a latent variable naive Bayes interpretation.

the operational component of the mixture, and given H the random variables $Z_1, \ldots, Z_D$ become independent (Fig. 3). Then, given $\mathbf{z}$, we can compute the likelihood using

$$\begin{aligned}\widehat{f}_{\boldsymbol{Z}}(\mathbf{z}) &= (\mathbf{b}_1^T\mathbf{A}_1 \circledast \cdots \circledast \mathbf{b}_D^T\mathbf{A}_D)^T\boldsymbol{\lambda} \\ &= (\circledast_{d=1}^D \mathbf{b}_d^T\mathbf{A}_d)\boldsymbol{\lambda}.\end{aligned}$$

The complexity of computing the likelihood of a data point $\mathbf{z}$ is $O(DKF)$. Tensor methods are commonly used to establish that the parameters of a generative model can be identified given higher order moments. This generative model theoretically has enough flexibility to capture highly complex distributions such as image manifolds. According to this model, a sample of the multivariate latent distribution can be generated by first drawing H according to p_H and then independently drawing samples for each variable Z_d from the conditional PDF $f_{Z_d|H}$.

1) Maximum Likelihood Estimation: The above analysis suggests fitting a low-rank CPD model on the D-way Fourier series coefficient tensor $\underline{\boldsymbol{\Phi}}$. To this end, we build a generative probabilistic model that assigns high probability to the transformed observed samples. We propose fitting the Fourier tensor coefficients indirectly on the latent space representation of the training data. Let us define matrices $\mathbf{B}_d \in \mathbb{C}^{(2K+1)\times M}$ as

$$\mathbf{B}_d(K+1+k_d, m) = e^{-j2\pi k_d \mathbf{z}_m(d)}.$$

Given M samples, we define the following NLL cost term

$$\begin{aligned}\mathcal{L}_{\text{NLL}} &:= -\frac{1}{M}\sum_{m=1}^{M}\log\Big(\hat{f}_{\boldsymbol{Z}}(\boldsymbol{z}_m)\Big) \\ &= -\frac{1}{M}\sum_{m=1}^{M}\log\Big((\circledast_{d=1}^D(\mathbf{B}_d(:,m)^T\mathbf{A}_d))\boldsymbol{\lambda}\Big),\end{aligned} \tag{13}$$

where $\boldsymbol{A}_d(K+1+k_d, h)$ holds $\Phi_{Z_d|H=h}[k_d]$, $\boldsymbol{\lambda}(f)$ holds $p_H(f)$. Note that we do not instantiate the full Fourier coefficient tensor but rather recover it implicitly, by minimizing the NLL term.

We can further restrict the model and reduce its learnable parameters by 50% by noticing that each column of the factor matrix $\mathbf{A}_d$ holds a valid characteristic function which is by definition conjugate symmetric around the origin, and equal to one at the origin, i.e.,

$$\begin{aligned}\mathbf{A}_d(K+1,:) &= \mathbf{1}^T, \text{ and} \\ \mathbf{A}_d(K+1+k,:) &= \mathbf{A}_d^*(K+1-k,:), \\ k \in [K], d &\in [D].\end{aligned}$$

Algorithm 1 CDE (Projected - SGD)

Input: $\mathbf{Z}, \mathbf{Z}_{\text{val}}, F, K, D, M_{\text{batch}}$
Initialize $\boldsymbol{\lambda}, \{\mathbf{A}_n\}_{n=1}^D, \boldsymbol{\theta}_g, \boldsymbol{\theta}_h$
repeat
  Sample M_{batch} data points
  Update network parameters via SGD
  for $d = 1$ **to** D **do**
    Update $\mathbf{A}_d$ via SGD
  end for
  Update $\boldsymbol{\lambda}$
  Project $\boldsymbol{\lambda}$ onto the probability simplex
  Compute $\mathcal{L}_{\text{NLL}} + \mathcal{L}_{\text{rec}}$ using $\mathbf{Z}_{\text{val}}$
until $\max_{\text{iter}}$ is reached or $\mathcal{L}_{\text{NLL}} + \mathcal{L}_{\text{rec}}$ stops diminishing

C. Optimization Procedure

By the above reasoning, instead of using decoupled two-stage training we suggest the following overall joint DR and density estimation optimization problem which we tackle by stochastic gradient descent

$$\begin{aligned}\min_{\boldsymbol{\theta}_h, \boldsymbol{\theta}_g, \{\mathbf{A}_d\}_{d=1}^D, \boldsymbol{\lambda}} & \frac{1}{M}\sum_{m=1}^{M}\Big(\|\mathbf{x}_m - g(h(\mathbf{x}_m;\boldsymbol{\theta}_h);\boldsymbol{\theta}_g)\|^2 - \\ & - \mu\log\Big((\circledast_{d=1}^D(\mathbf{B}_d(:,m)^T\mathbf{A}_d))\boldsymbol{\lambda}\Big)\Big) + \sum_{d=1}^{D}\rho\|\mathbf{A}_d\|_F^2 \\ \text{s.t. } & \boldsymbol{\lambda} \geq \mathbf{0}, \mathbf{1}^T\boldsymbol{\lambda} = 1, \\ & \mathbf{A}_d(K+1,:) = \mathbf{1}^T, \\ & \mathbf{A}_d(K+1+k,:) = \mathbf{A}_d^*(K+1-k,:).\end{aligned}$$

The optimization criterion that guides CDE consists of three terms: the reconstruction loss of the auto-encoder, NLL of the density estimation component, and Frobenius norm regularization. In the above formulation, $\mu \geq 0$ is a regularization parameter which balances the reconstruction error versus the maximum likelihood estimation. The number of coefficients K controls the desired smoothness of the joint density, while the number of latent dimensions D and the rank F control the expressivity.

We refer to this approach as Compressed-domain Density Estimation with Hidden Tensor Factorization (CDE-HTF). Figure (1) presents the network structure corresponding to the final joint problem formulation. We solve the proposed optimization problem using projected Stochastic Gradient Descent (SGD). We initialize the Fourier tensor-related parameters using random initialization, while for $\boldsymbol{\theta}_g$ and $\boldsymbol{\theta}_h$, it was empirically observed that auto-encoder pre-training was most effective. At each step we update $\boldsymbol{\theta}_g$, $\boldsymbol{\theta}_h$, factors $\mathbf{A}_d$ and $\boldsymbol{\lambda}$ simultaneously by first sampling a batch of size M_{batch} and taking a gradient step. After that, we project $\boldsymbol{\lambda}$ to the probability simplex. For the termination of the algorithm we compute the cost function on a validation set and stop if a number of maximum iterations has been reached or the log-likelihood has not improved in the last T iterations. The full procedure is shown in Algorithm 1.

IV. Experimental Results

In this section, we evaluate the proposed approach using various datasets and evaluation criteria, ranging from sampling

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2022.3158422

of toy 3-D examples to real MNIST and Fashion-MNIST images, and regression and anomaly detection tasks using standard tabular datasets from the UCI database. We compare with density estimation and anomaly detection baselines from the deep learning literature, including standard VAEs, Real-NVP, MAF, MADE and GF for reference.

A. Toy Datasets

We begin with modeling the joint density function of a subset of MNIST images, consisting of only 0s and 8s using the proposed CDE model. For these experiments the network architecture considered is a four-layer network encoder of 784, 128, 64, 32, neurons respectively (with the decoder being a mirrored version of the encoder), and ReLU activation functions. In Figure 2, we visualize random samples learned by the proposed model for different values of F and K to show that only a few parameters are needed to obtain a model that is flexible enough to fit the distribution in great detail. The first row represents results for fixed $F = 4$ and different values of $K \in [1, 3, 5]$, while the second row represents results for fixed $K = 3$ and different values of $F \in [2, 4, 8]$. Increasing K generates sharper digits, while increasing F better differentiates the samples of the two digits.

We then continue with modeling three toy 3-D datasets, namely Swiss-roll, S-curve, and Fish-bowl where we are given 3000 training data points from each dataset and we use CDE to randomly sample 5000 synthetic data points. For all the datasets considered, the auto-encoder structure we use is FC (3, 128, ReLU), FC (128, 64, ReLU), FC (64, 32, ReLU), FC (32, 2, none), FC (2, 32, ReLU), FC (32, 64, ReLU), FC (64, 128, ReLU), FC (128, 3, none), and the tensor parameters are set to $(K, F) = (5, 10)$. We provide an illustration of the 2-D latent space learned via latent synthetic samples drawn using the proposed method. Using the approximate inverse map of the decoder, we can back-transform latent samples into samples in the original data space and visualize the learned distribution in the original space. The results in Figure 4 showcase that the proposed framework is capable of learning the structure of the data, notably in critical regions where the curvature is very high – which is interesting.

B. Tabular Datasets

Next, we evaluate the proposed approach on several regression tasks using tabular datasets described in Table I. We compare our approach against standard baselines. For each dataset we split the data in two subsets: 80% used for training and 20% used for testing. The parameters for each method are chosen using 5-fold cross-validation and we report the Mean Absolute Error (MAE) for the unseen data samples. The results underline the superior performance of the proposed method for inference tasks. Regarding the auto-encoder's parameters for the datasets, the number of hidden layers varies according to the dataset dimensionality – from three (MINIBOONE dataset) to six (ISOLET dataset). The most critical one is the hidden layer dimensionality $D \in \{16, 32, 48, 64\}$, and concerning tensor parameters, the important ones are the tensor rank $F \in \{5, 10, 20, 30, 50\}$ and the smoothing parameter $K \in \{5, 10, 15, 20, 30\}$. The learning, drop-out rates, and regularization parameters were sampled from a uniform distribution in the range $[0.05, 0.2]$. That is, we randomly sampled parameters from this range using a uniform distribution and used cross-validation to select the best set of parameters. The initial weight matrices were all sampled from the uniform distribution within the range $[-1, 1]$.

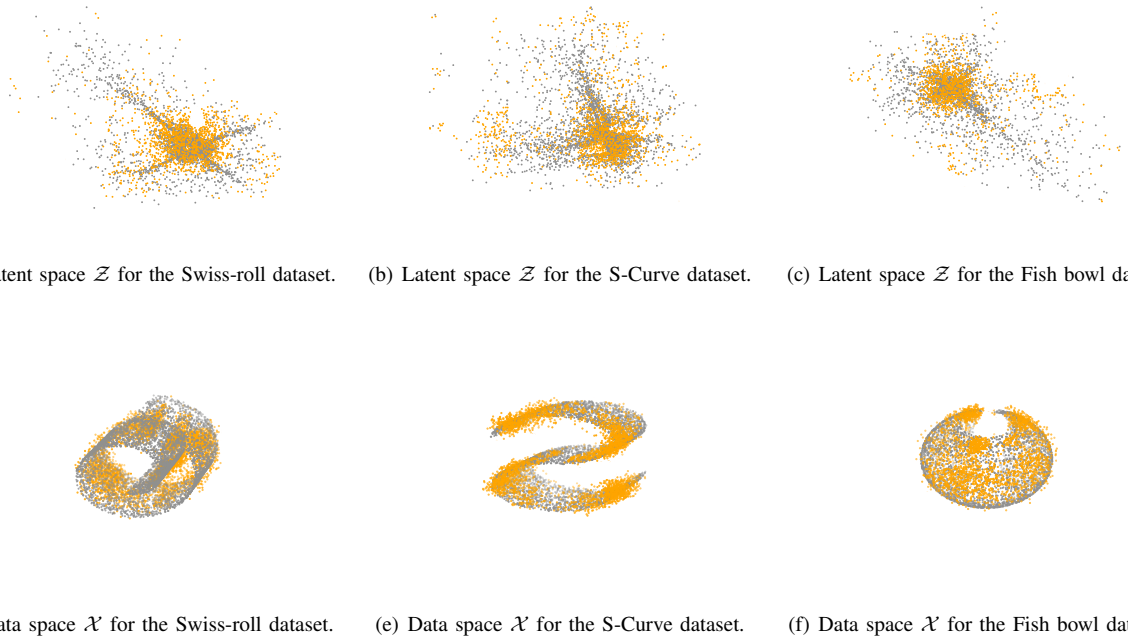
We used Adam optimizer [37] with a batch size of 500. Overall, we observe that CDE outperforms the baselines on almost all datasets, and performs comparable to the winning method in the remaining ones. More specifically, CDE shows significantly lower test-set MAE on MINIBOONE, Gas Sensor, and BlogFeedback dataset compared to Real-NVP and MAF, which appear to be the next best performing models. Additionally CDE has a clear lead against the VAE especially on Gas Sensor, IDA2016Challenge, and ISOLET dataset, which confirms our initial motivation of using a non-parametric latent density estimator to improve model flexibility.

C. Image Datasets

We consider grayscale images from the MNIST and Fashion-MNIST database, which both contain a set of $60,000$ training observations of 28×28 pixels ($N = 784$) from 10 classes. Regarding MNIST, the most critical parameters include the encoder architecture, which consists of four hidden layers of 784, 128, 64, 32, neurons respectively (the decoder network has a mirrored structure), ReLU activation function, tensor rank which is fixed to $F = 50$, smoothing parameter $K = 5$, and the learning rate which is fixed to $\alpha = 0.0001$. For Fashion-MNIST, the model parameters are fixed to be the same as for the MNIST dataset, with the encoder network $(784, 256, 128, 64, 32, 16)$ being the only exception. See also Figure 7, which shows the distribution of the components of the latent variable H after training the generative model. These bar-plots tell us that the rank of the compressed density model is essentially $F = 30$, but for exploratory modeling purposes, we set $F = 50$ (the parameter F is actually an upper bound on tensor rank) and encourage sparsity of the latent components through our optimization problem formulation. We sample from the learned lower-dimensional latent joint generative model ($D = 32$ for MNIST and $D = 16$ for Fashion-MNIST) – See Section III-B for the detailed sampling process – and provide visualization of the generated data. The resulting 100 randomly drawn samples, which are impressively more pleasing to the eye in direct comparison with other well-known models such as MADE and GFs are shown in Figures 5 and 6.

We note that in our case, no prior is imposed on the latent variables, so we do not have issues such as "posterior collapse" which has been observed to occur in VAEs. Degeneration ("collapse") could occur in principle for our model if the joint characteristic function is reduced to a rank-1 tensor during training. An easy way to check this would be through the mixture distribution of the latent components, i.e., the probability mass function of the latent variable H, $p_H(f)$. If after training the generative model only one element of $p_H(\cdot)$ is nonzero, i.e., only one component "survives", then the characteristic tensor is rank-1, and degeneration / collapse

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2022.3158422



(a) Latent space $\mathcal{Z}$ for the Swiss-roll dataset. (b) Latent space $\mathcal{Z}$ for the S-Curve dataset. (c) Latent space $\mathcal{Z}$ for the Fish bowl dataset.

(d) Data space $\mathcal{X}$ for the Swiss-roll dataset. (e) Data space $\mathcal{X}$ for the S-Curve dataset. (f) Data space $\mathcal{X}$ for the Fish bowl dataset.

Fig. 4: Three toy 3-D (gray data-points) datasets and corresponding samples drawn via CDE. CDE maps raw samples to latent features through a mapping $\boldsymbol{h} : \mathbb{R}^3 \to \mathbb{R}^2$ that is learned by an auto-encoder. The non-parametric latent density model allows efficient sample generation in the latent space (orange data-points on the images of the first row). The approximate inverse map of the decoder, back-transforms latent samples into samples in the data space (orange data-points on the images of the second row. See text for further details.

Data set	N	M	VAE	Real-NVP	MAF	GF	CDE
MINIBOONE	51	130065	3.69 ± 0.68	3.18 ± 0.16	3.17 ± 0.45	$\mathbf{3.15 \pm 0.52}$	$\mathbf{3.12 \pm 0.43}$
BSDS300	63	50000	0.37 ± 0.08	0.60 ± 0.02	$\mathbf{0.32 \pm 0.03}$	0.48 ± 0.03	$\mathbf{0.30 \pm 0.02}$
Gas Sensor	128	13910	1.46 ± 0.04	1.31 ± 0.31	1.23 ± 0.44	1.23 ± 0.30	$\mathbf{1.21 \pm 0.84}$
Musk	168	6598	0.40 ± 0.51	0.22 ± 0.68	$\mathbf{0.12 \pm 0.07}$	0.19 ± 0.08	$\mathbf{0.13 \pm 0.23}$
IDA2016Challenge	171	76000	0.23 ± 0.06	0.18 ± 0.09	$\mathbf{0.12 \pm 0.05}$	$\mathbf{0.10 \pm 0.09}$	$\mathbf{0.11 \pm 0.16}$
BlogFeedback	281	60021	2.43 ± 0.22	$\mathbf{2.35 \pm 0.21}$	2.37 ± 0.22	2.37 ± 0.42	$\mathbf{2.32 \pm 0.17}$
ISOLET	617	7797	1.41 ± 0.31	$\mathbf{1.09 \pm 0.04}$	1.11 ± 0.38	1.85 ± 0.21	$\mathbf{1.03 \pm 0.56}$

TABLE I: Dataset information and test-set MAE on UCI datasets. For each test sample, we choose the response variable Y at random and estimate using Stochastic Gradient Ascent.

to a separable latent space distribution occurs. Throughout our experiments, although the datasets can be well approximated with lower ranks, such degeneration did not take place. We show this in Figure 7, where we demonstrate the distribution of the latent variable H after training the generative model on MNIST (left figure) and Fashion-MNIST (right figure).

Although at first glance, the images generated by VAE have cleaner and thicker strokes, they are also more blurry and distorted than those produced by CDE. The Fashion-MNIST data help bring this out more clearly: one can see that CDE allows capturing and representing more details in the items, while the samples drawn from the VAE are much more blurry. The overall conclusion from the experiments is that while the quality of our synthetic images is competitive against the VAE and considerably better than that of the samples generated by the rest of the models considered, the proposed framework is superior for regression tasks.

Because the dimensionality of the latent space is an important factor in the architecture, we demonstrate sampling results for different dimensionalities on Fashion-MNIST. We repeat the experiment by setting $F = 50$ and $K = 5$. We randomly sample from the learned lower-dimensional latent joint generative model for $D = 6, 10, 14$ and provide visualization of the generated data in the input space. The resulting 100 randomly drawn samples are shown in Figure 8. We can see that increasing latent dimenionality from $D = 6$ to $D = 14$ significantly improves sample quality. Results for $D = 16$ can be seen on Figure 6 which yield high-quality image samples. Larger latent dimensionality does not further improve the samples (and eventually degrades it, as expected) so we opt for the smallest dimensionality that yields good results.

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2022.3158422

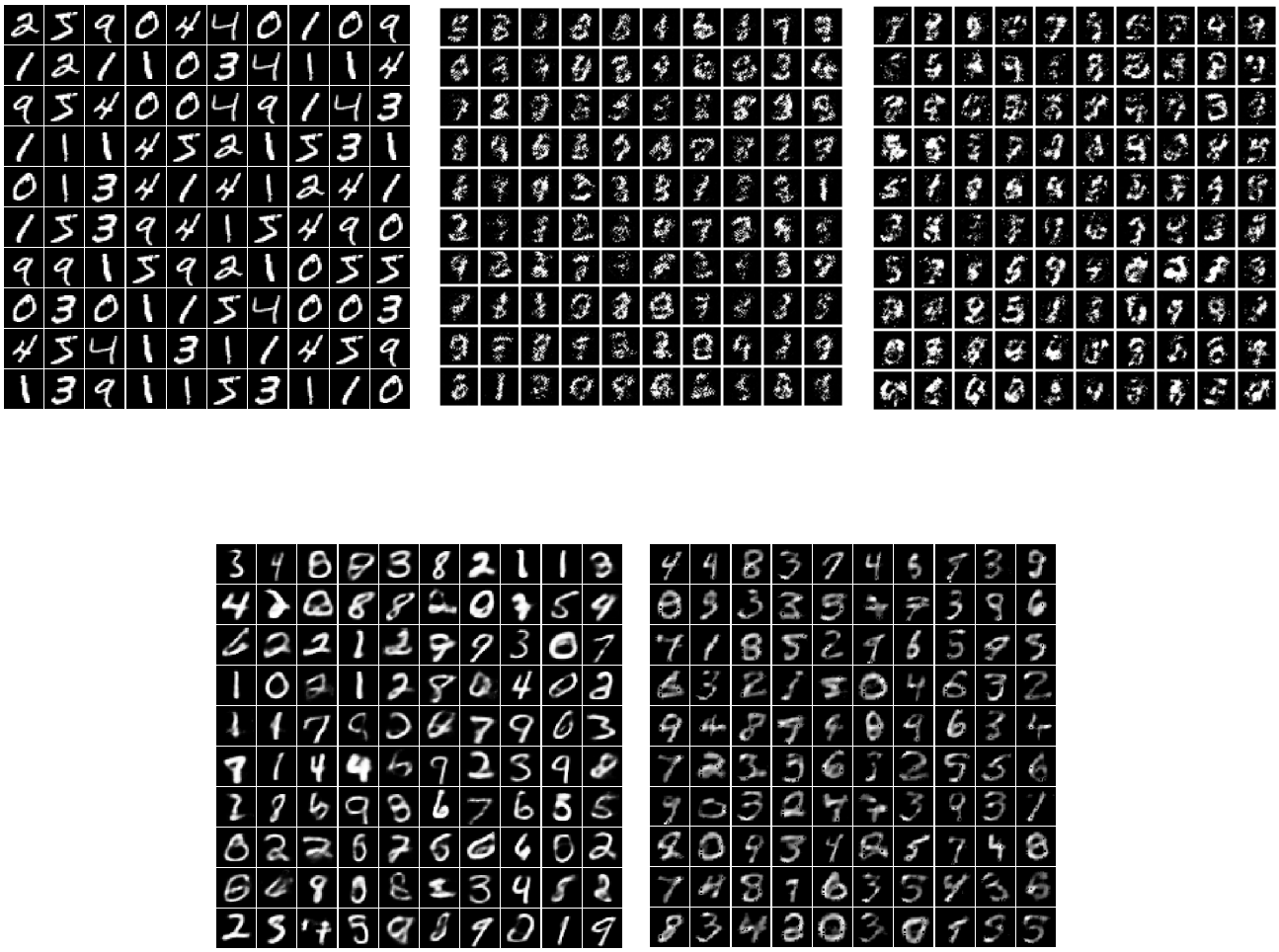


Fig. 5: Synthetic samples drawn from the joint density of MNIST from various models. From left to right: Ground Truth, Masked auto-encoder for Distribution Estimation (MADE), Gaussianization Flows (GF), Variational auto-encoder (VAE), **Proposed: CDE**.

D. Anomaly Detection Using Real Data

We use four public datasets[‡]: KDDCUP99, Thyroid, Arrhythmia, and KDDCUP-Rev. The (instance number M, dimension N, anomaly ratio (%)) of each dataset is (494021, 121, 20), (3772, 6, 2.5), (452, 274, 15), and (121597, 121, 20). For categorical features, we further used one-hot representation to encode them. Regarding Thyroid, there are three classes in the original dataset. We treat the hyperfunction class as the anomaly class and the other two classes are treated as normal class. Regarding Arrhythmia, the smallest classes, including $3, 4, 5, 7, 8, 9, 14$, and 15, are combined to form the anomaly class, and the rest of the classes are combined to form the normal class. We randomly extracted 50% of the data and assigned it to the training subset and the rest to the testing subset. In our experimental setting for anomaly detection, clean training data is adopted – that is, during the training, only normal data were used. We assume that the percentage of anomalous data points is known, and our goal is to detect which data points in the testing subset are most likely to be outliers. Towards this end, at the testing stage the likelihood of each testing sample in the compressed domain was evaluated and sorted in order to detect the anomalies as the points with the smallest likelihood. By knowing the percentage of anomalies, we can indicate the exact number of outliers and output the data samples with the smallest likelihood values.

The network structures of CDE used for individual datasets are summarized as follows.

- For KDDCUP, the auto-encoder network runs with FC (120, 60, tanh), FC (60, 30, tanh), FC (30, 20, tanh), FC (20, 10, none), FC (10, 20, tanh), FC (20, 30, tanh), FC (30, 60, tanh), FC (60, 120, none).
- The auto-encoder network for Thyroid runs with FC (6, 12, tanh), FC (12, 4, tanh), FC (4, 2, none), FC (2, 4,

[‡]Datasets can be downloaded at https://kdd.ics.uci.edu/ and http://odds.cs.stonybrook.edu.

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2022.3158422

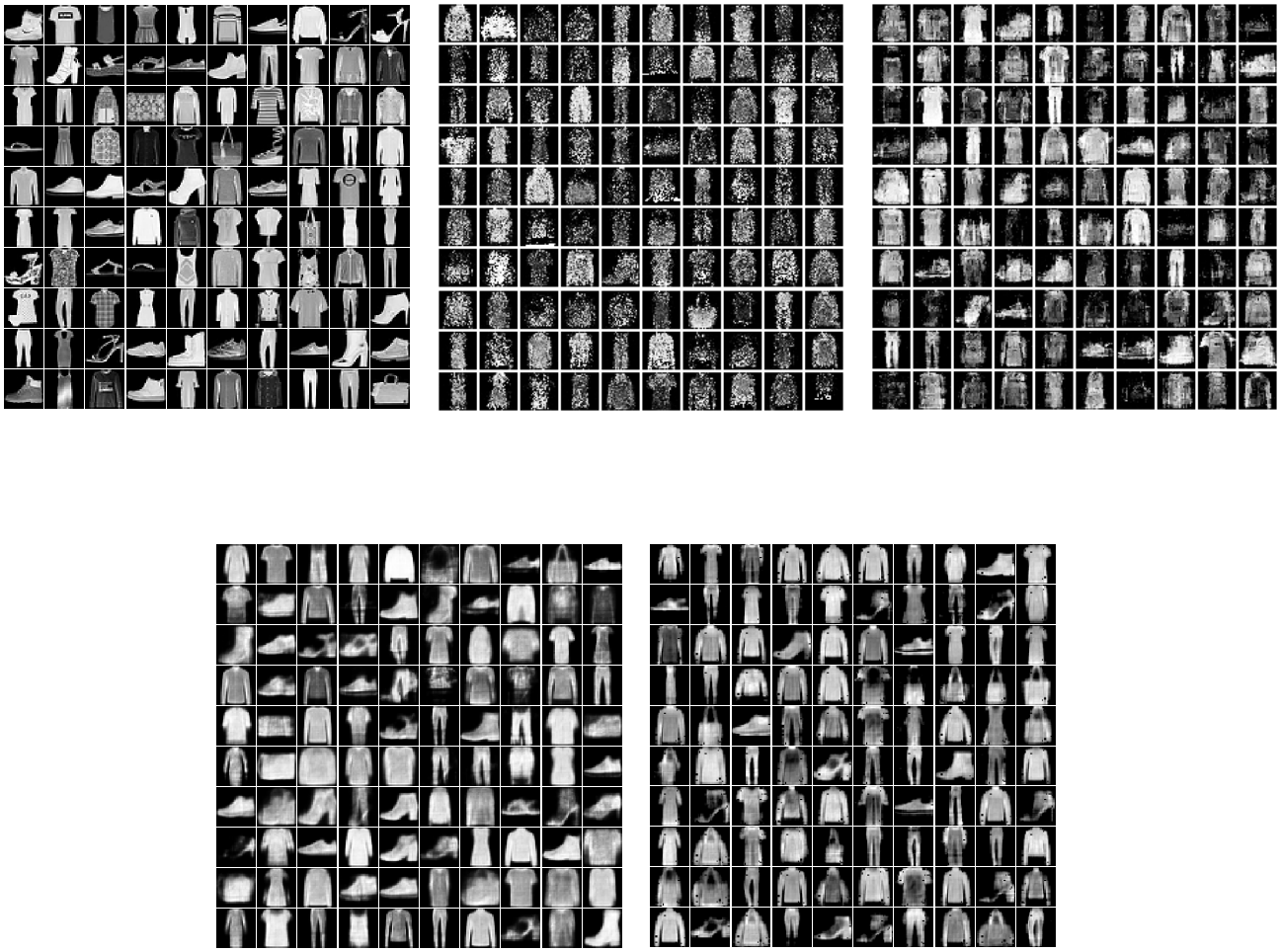


Fig. 6: Synthetic samples drawn from the joint density of Fashion-MNIST from various models. From left to right: Ground Truth, Masked auto-encoder for Distribution Estimation (MADE), Gaussianization Flows (GF), Variational auto-encoder (VAE), **Proposed: CDE**.

tanh), FC (4, 12, tanh), FC (12, 6, none).

- The auto-encoder network for Arrhythmia runs with FC (274, 64, tanh), FC (64, 32, none), FC (32, 64, tanh), FC (64, 274, none).
- The auto-encoder network for KDDCUP-Rev runs with FC (120, 60, tanh), FC (60, 30, tanh), FC (30, 20, tanh), FC (20, 10, none), FC (10, 20, tanh), FC (20, 30, tanh), FC (30, 60, tanh), FC (60, 120, none).

As metrics, precision, recall, and F1 score are calculated. We run experiments 20 times for each dataset split by 20 different random seeds. Table II reports the average scores and standard deviations (in brackets). Compared to the baselines considered, CDE achieves the highest performance – CDE is superior to both VAE and DAGMM for each evaluation criterion except for precision on the Arrhythmia dataset. This result suggests that our proposed latent nonparametric density estimation approach can provide more expressive models which can bring better performance in important detection tasks as well.

V. Conclusions

In this work, we introduced Compressed Density Estimation (CDE), a novel probabilistic latent density model that builds upon deep auto-encoder networks and non-parametric multivariate density modeling in the Fourier domain. We propose using an auto-encoder to embed the data into a latent code space by minimizing reconstruction error, and a regularization over the latent space which maximizes the likelihood of the hidden code vector and is modelled using a low-rank characteristic tensor approach.

We investigated whether leveraging probabilistic (non-parametric) low-rank tensor models in the Fourier domain as a latent distribution model can improve the expressivity of density models. By jointly optimizing the auto-encoder and the latent density model, we can better capture the latent distribution of

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2022.3158422

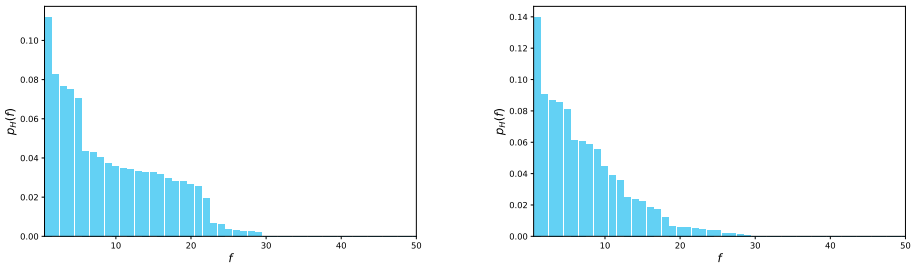


Fig. 7: Distribution of the components of the latent variable H after training the generative model on MNIST (left figure) and Fashion-MNIST (right figure).

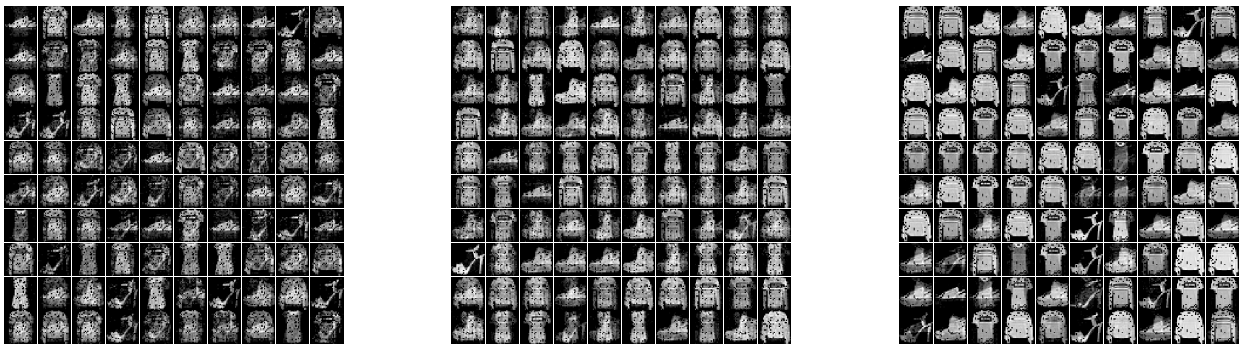


Fig. 8: Fashion-MNIST samples for different values of latent dimensionality ($D = 6, D = 10, D = 14$).

Dataset	Methods	Precision	Recall	F1
KDDCup	VAE	0.9524 (0.0047)	0.9140 (0.0052)	0.9326 (0.0052)
	DAGMM	0.9427 (0.0055)	0.9578 (0.0051)	0.9507 (0.0052)
	CDE	**0.9565 (0.0046)**	**0.9712 (0.0048)**	**0.9641 (0.0045)**
Thyroid	VAE	**0.6575 (0.0371)**	0.5743 (0.0583)	0.6357 (0.0583)
	DAGMM	0.4658 (0.0481)	0.4902 (0.0452)	0.4752 (0.0497)
	CDE	0.6560 (0.0572)	**0.6740 (0.0493)**	**0.6703 (0.0592)**
Arrythmia	VAE	0.4375 (0.0538)	0.4340 (0.0496)	0.4302 (0.0482)
	DAGMM	**0.5358 (0.0468)**	**0.5592 (0.0475)**	**0.5403 (0.0421)**
	CDE	0.5299 (0.0400)	0.5551 (0.0418)	0.5389 (0.0420)
KDDCup-rev	VAE	0.9771 (0.0058)	0.9779 (0.0004)	0.9678 (0.0018)
	DAGMM	0.9762 (0.0038)	0.9823 (0.0017)	0.9709 (0.0021)
	CDE	**0.9866 (0.0008)**	**0.9872 (0.0015)**	**0.9871 (0.0012)**

TABLE II: Average (over 20 runs) and standard deviations (in brackets) of Precision, Recall and F1 score.

data representations obtained by the auto-encoder. Experimental results demonstrated the effectiveness of the proposed joint optimization approach, which is able to learn complex high dimensional distributions using a parsimonious model with few tuning parameters.

REFERENCES

[1] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 215–10 224.

[2] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[3] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," *arXiv preprint arXiv:1511.06349*, 2015.

[4] 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, pp. 1–19.

[5] B. W. Silverman, *Density estimation for statistics and data analysis*. CRC press, 1986, vol. 26.

[6] G. J. McLachlan and D. Peel, *Finite mixture models*. John Wiley & Sons, 2004.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[8] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations*, 2014.

[9] B. Dai and D. Wipf, "Diagnosing and enhancing VAE models," *arXiv preprint arXiv:1903.05789*, 2019.

[10] M. Rosca, B. Lakshminarayanan, and S. Mohamed, "Distribution matching in variational inference," *arXiv preprint arXiv:1802.06847*,

2018.
[11] A. V. Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *International Conference on Machine Learning*, vol. 48, 20–22 Jun 2016, pp. 1747–1756.
[12] L. Dinh, D. Krueger, and Y. Bengio, "NICE: Non-linear independent components estimation," *arXiv preprint arXiv:1410.8516*, 2015.
[13] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," in *International Conference on Learning Representations*, 2016.
[14] J. Ho, X. Chen, A. Srinivas, Y. Duan, and P. Abbeel, "Flow++: Improving flow-based generative models with variational dequantization and architecture design," in *International Conference on Machine Learning*, 2019, pp. 2722–2730.
[15] M. Amiridi, N. Kargas, and N. D. Sidiropoulos, "Nonparametric multivariate density estimation: A low-rank characteristic function approach," *arXiv preprint arXiv:2008.12315*, 2020.
[16] R. A. Harshman *et al.*, "Foundations of the PARAFAC procedure: Models and conditions for an explanatory multimodal factor analysis," pp. 1–84, 1970.
[17] Y. LeCun, "The MNIST database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.
[18] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.
[19] D. W. Scott, "Feasibility of multivariate density estimates," *Biometrika*, vol. 78, no. 1, pp. 197–205, 1991.
[20] M. D. Hoffman and M. J. Johnson, "Elbo surgery: yet another way to carve up the variational evidence lower bound," in *Workshop in Advances in Approximate Bayesian Inference, NIPS*, vol. 1, 2016, p. 2.
[21] B. Uria, I. Murray, and H. Larochelle, "RNADE: The real-valued neural autoregressive density-estimator," in *Advances in Neural Information Processing Systems*, 2013, pp. 2175–2183.
[22] M. Germain, K. Gregor, I. Murray, and H. Larochelle, "MADE: Masked autoencoder for distribution estimation," in *International Conference on Machine Learning*, 2015, pp. 881–889.
[23] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *International Conference on Machine Learning*, vol. 37, 2015, pp. 1530–1538.
[24] G. Papamakarios, T. Pavlakou, and I. Murray, "Masked autoregressive flow for density estimation," in *Advances in Neural Information Processing Systems*, 2017, pp. 2338–2347.
[25] C. Meng, Y. Song, J. Song, and S. Ermon, "Gaussianization flows," in *International Conference on Artificial Intelligence and Statistics*, 2020, pp. 4336–4345.
[26] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *Journal of Mathematics and Physics*, vol. 6, no. 1-4, pp. 164–189, 1927.
[27] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.
[28] L. Cheng, X. Tong, S. Wang, Y.-C. Wu, and H. V. Poor, "Learning nonnegative factors from tensor data: Probabilistic modeling and inference algorithm," *IEEE Transactions on Signal Processing*, vol. 68, pp. 1792–1806, 2020.
[29] L. Xu, L. Cheng, N. Wong, and Y.-C. Wu, "Overfitting avoidance in tensor train factorization and completion: Prior analysis and inference," in *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2021, pp. 1439–1444.
[30] N. Kargas, N. D. Sidiropoulos, and X. Fu, "Tensors, learning, and "Kolmogorov extension" for finite-alphabet random vectors," *IEEE Transactions on Signal Processing*, vol. 66, no. 18, pp. 4854–4868, 2018.
[31] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010.
[32] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
[34] G. Plonka, D. Potts, G. Steidl, and M. Tasche, *Numerical Fourier Analysis*. Springer, 2018.
[35] J. C. Mason, "Near-best multivariate approximation by fourier series, chebyshev series and chebyshev interpolation," *Journal of Approximation Theory*, vol. 28, no. 4, pp. 349–358, 1980.
[36] D. C. Handscomb, *Methods of numerical approximation: lectures delivered at a Summer School held at Oxford University, September 1965*. Elsevier, 2014.
[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *3rd International Conference on Learning Representations, ICLR*, 2015.

Magda Amiridi received the Diploma in Electrical and Computer Engineering from the Technical University of Crete (TUC), Chania, Greece, in 2018. Currently, she is working towards the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Virginia, where she is also affiliated with the Signal and Tensor Analytics Research (STAR) group under the supervision of Professor N. D. Sidiropoulos. She is working on theoretical foundations and algorithmic approaches for non-parametric probabilistic modeling using low-rank tensors. She is a student member of the IEEE and student chapter of IEEE's Signal Processing Society at UVA.

Nikos Kargas received the Diploma and M.Sc. degree in Electronic and Computer engineering from the Technical University of Crete (TUC), Chania, Greece, in 2013 and 2015, respectively, and Ph.D. degree in Electrical Engineering from University of Minnesota, Minneapolis, in 2020. His interests are in the areas of Machine learning, Statistics and Optimization. A major focus of his work is on tensor methods for high-dimensional distribution and general nonlinear function learning. His recent research topics include non-parametric density estimation, data completion/regression and time-series forecasting.

Nicholas D. Sidiropoulos (Fellow, IEEE) received the Diploma in Electrical Engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece, and the M.S. and Ph.D. degrees in Electrical Engineering from the University of Maryland at College Park, College Park, MD, USA, in 1988, 1990, and 1992, respectively. He has served on the faculty of the University of Virginia, the University of Minnesota, and the Technical University of Crete, Greece, prior to his current appointment as Louis T. Rader Professor of ECE at UVA. From 2015 to 2017, he was an ADC Chair Professor with the University of Minnesota. His research interests are in signal processing, communications, optimization, tensor decomposition, and factor analysis, with applications in machine learning and communications. He received the NSF/CAREER award in 1998, the IEEE Signal Processing Society (SPS) Best Paper Award in 2001, 2007, and 2011, served as IEEE SPS Distinguished Lecturer (2008–2009), and as Vice President - Membership of IEEE SPS (2017–2019). He received the 2010 IEEE Signal Processing Society Meritorious Service Award, and the 2013 Distinguished Alumni Award from the University of Maryland, Department of ECE. He is a fellow of EURASIP (2014).