MACHINE
LEARNING
Science and Technology

CrossMark

OPEN ACCESS

RECEIVED
23 March 2023

REVISED
6 June 2023

ACCEPTED FOR PUBLICATION
13 July 2023

PUBLISHED
11 August 2023

PAPER

Anomaly detection in aeronautics data with quantum-compatible discrete deep generative model

Thomas Templin[1,*], Milad Memarzadeh[2], Walter Vinci[3], P Aaron Lott[4], Ata Akbari Asanjan[2], Anthony Alexiades Armenakas[4,5] and Eleanor Rieffel[6]

[1] Data Sciences Group, NASA Ames Research Center, Moffett Field, CA 94035, United States of America
[2] Universities Space Research Association, Data Sciences Group, NASA Ames Research Center, Moffett Field, CA 94035, United States of America
[3] HP SCDS, 24009 León, Spain
[4] Universities Space Research Association, Quantum Artificial Intelligence Laboratory, NASA Ames Research Center, Moffett Field, CA 94035, United States of America
[5] Department of Physics, Harvard University, Cambridge, MA 02138, United States of America
[6] Quantum Artificial Intelligence Laboratory, NASA Ames Research Center, Moffett Field, CA 94035, United States of America
[*] Author to whom any correspondence should be addressed.

E-mail: thomas.templin@nasa.gov

Keywords: generative modeling, deep learning, variational autoencoder, anomaly detection, restricted Boltzmann machine, Gibbs sampling, quantum-assisted machine learning

Supplementary material for this article is available online

Abstract

Deep generative learning cannot only be used for generating new data with statistical characteristics derived from input data but also for anomaly detection, by separating nominal and anomalous instances based on their reconstruction quality. In this paper, we explore the performance of three unsupervised deep generative models—variational autoencoders (VAEs) with Gaussian, Bernoulli, and Boltzmann priors—in detecting anomalies in multivariate time series of commercial-flight operations. We created two VAE models with discrete latent variables (DVAEs), one with a factorized Bernoulli prior and one with a restricted Boltzmann machine (RBM) with novel positive-phase architecture as prior, because of the demand for discrete-variable models in machine-learning applications and because the integration of quantum devices based on two-level quantum systems requires such models. To the best of our knowledge, our work is the first that applies DVAE models to anomaly-detection tasks in the aerospace field. The DVAE with RBM prior, using a relatively simple—and classically or quantum-mechanically enhanceable—sampling technique for the evolution of the RBM's negative phase, performed better in detecting anomalies than the Bernoulli DVAE and on par with the Gaussian model, which has a continuous latent space. The transfer of a model to an unseen dataset with the same anomaly but without re-tuning of hyperparameters or re-training noticeably impaired anomaly-detection performance, but performance could be improved by post-training on the new dataset. The RBM model was robust to change of anomaly type and phase of flight during which the anomaly occurred. Our studies demonstrate the competitiveness of a discrete deep generative model with its Gaussian counterpart on anomaly-detection problems. Moreover, the DVAE model with RBM prior can be easily integrated with quantum sampling by outsourcing its generative process to measurements of quantum states obtained from a quantum annealer or gate-model device.

1. Introduction

The field of machine learning has experienced an explosion in the development of deep-learning methods at the beginning of the 21st century, due to the flexibility, scalability, and superior performance of deep learning in classification, prediction, data generation, anomaly detection, and other applications [1–4]. The phenomenal success of deep learning, which refers to machine-learning techniques that use artificial-neural-network models with many layers, has been enabled by the widespread availability of specialized graphics processing units (GPUs) to perform computing-intensive linear-algebra operations on vectors, matrices, and tensors.

One way to characterize the training process of a neural network is to differentiate between supervised and unsupervised learning [5, 6]. In supervised learning, a 'teacher' imposes a set of desired input-output relationships on the network. For example, the training set might contain an extra column that specifies the desired output of the network, such as a class label. The class label or other supervisory information is not available during testing, when the performance of the network is evaluated. In unsupervised learning, no such oversight is provided, and the network's response is self-organized and solely relies on the interplay between external input, intrinsic connectivity, network dynamics, and the value of a cost function that the network attempts to minimize. Unsupervised learning is computationally more complex than supervised learning and still a largely unresolved problem in machine learning. It has attracted considerable research effort [2, 7, 8] because it holds the potential to uncover the statistical structure and hidden correlations of large unlabeled datasets, which constitute the predominant form of today's data.

Generative modeling is a widely used machine-learning technique that attempts to estimate the probability distribution of a dataset and use it to generate new data. Deep generative models such as generative adversarial networks (GANs) [9], variational autoencoders (VAEs) [10], and deep belief networks [11] have been extensively applied to machine-learning use cases in science and engineering. GANs do not explicitly define a density function or approximate likelihood but define the probability density implicitly, by producing samples from it [9, 12]. GANs are known for generating realistic high-resolution images and have also been deployed, with increasing popularity, to anomaly-detection tasks [13–15]. However, GANs are difficult to train due to instability issues, such as non-convergence, posterior-mode collapse, and vanishing or exploding gradients [12, 16–19]. In the studies reported in this paper, we use VAEs for generative modeling because of their greater training stability and efficient inference mechanism. VAEs employ the evidence lower bound (ELBO) as a variational lower bound on the exact log likelihood, incorporate regularization via a prior, and allow estimation of the log likelihood via importance sampling [10, 20]. The (negative) ELBO is a well-defined, fully differentiable loss function whose gradients are used to efficiently optimize network weights through backpropagation, permitting competitive performance in mining large datasets. Furthermore, the β-VAE, used in our experiments, allows a weighting of the autoencoding (reconstruction) and Kullback-Leibler (KL)-divergence terms in the variational ELBO objective [21, 22].

Because of the widespread applicability of the normal distribution owing to the central limit theorem and the difficulty of propagating gradients through discrete variables, the majority of VAE and other generative-model designs reported in the literature use the continuous Gaussian distribution to model the prior and approximate the posterior distribution of the latent variables given input data. However, many deep-learning use cases rely on discrete latent variables to represent the required distributions, such as in applications in supervised and unsupervised learning, attention models, language models, and reinforcement learning [23–25]. In particular, if the values of latent variables are to be computed by quantum computers, the latent variables need to be discrete because projective qubit measurements in the computational basis produce eigenvalues of -1 or $+1$. See supplementary section S1 for a more in-depth account of the importance of discrete-variable models.

Discrete VAEs (DVAEs) and quantum VAEs have been used to generate new data from samples from the VAE's latent space after the VAE was trained on a dataset such as MNIST or Omniglot, and the quality of data generation (fit of the VAE's model distribution to the distribution of the input data) was assessed by estimating the log likelihood of test data [26–32]. In addition, the application of VAEs to anomaly detection has become increasingly popular in recent years. An and Cho [33] suggested an anomaly-detection method in which the anomaly score of a VAE is used as a Monte Carlo estimate of the reconstruction log likelihood (called 'reconstruction probability' in the paper). Haowen Xu *et al* [34] used a VAE for the detection of anomalies in univariate time series, preprocessed with sliding time windows, representing seasonal key performance indicators in web applications. Several studies have incorporated recurrent neural networks (RNNs) into VAEs by equipping the VAE's encoder and decoder with long short-term memory (LSTM) constructs [35–39]. The LSTM-VAE approach was also applied to anomaly detection in telemetry data from the Soil Moisture Active Passive (SMAP) satellite and the Mars Curiosity rover [40]. However, the training of a VAE equipped with an RNN architecture on multidimensional time series is computationally costly and may overlook local temporal dependencies. To remedy these shortcomings, Memarzadeh *et al* [41] designed a convolutional VAE (CVAE) to detect anomalies in multivariate time series of commercial flights, a task on which the model achieved state-of-the-art performance.

VAEs and β-VAEs have also been applied to other problems in aerospace. For example, Yang *et al* [42] used a VAE in an inverse-design-optimization framework to learn the pressure distribution over a wind-turbine airfoil and then sampled from the VAE's latent space to generate highly realistic artificial inputs to a feedforward neural network that, in turn, predicted aerodynamic variables and shape parameters of the airfoil. Kang *et al* [43] designed β-VAEs (with various β values) to produce physically informative latent spaces, to be used in reduced-order modeling. The higher the value of β, the greater was the latent space's compression, without loss of information, and the two informative latent variables of the model with the highest β value coincided with the causal factors of the training dataset, Mach number and angle of attack.

For the studies reported in this paper, we developed convolutional VAEs with continuous (Gaussian) and discrete (Bernoulli and Boltzmann) priors to detect anomalies in multivariate time-series data of commercial flights. The Boltzmann prior is implemented as a restricted Boltzmann machine (RBM) [44]. The VAE with Gaussian prior and the RBM network of the VAE model with RBM prior are derivations of the CVAE model presented in [41] and of the DVAE model depicted in [31], respectively. Whereas the CVAE model presented in [41] employed multiple encoders and decoders and input data of highly correlated features were directed through separate encoders and decoders, the VAEs used in the studies reported in this paper possess a single unified encoder and decoder (see appendix G). Also, in the latent-space RBM-prior network used in [31], positive-phase samples (latent variables of the DVAE) were split into two equally sized parts ('visible' and 'hidden' units) to calculate positive-phase energies. By contrast, the positive phase of the RBM network used in the studies reported here contains a true hidden layer that is evolved from the DVAE's posterior latents by one-step Gibbs sampling. The contrasting RBM architectures are described in detail in section 3.4 and shown in figure 2.

The contributions and motivations of this work are as follows. We developed DVAE models with a factorized Bernoulli prior and with an RBM prior with a novel positive-phase architecture of the RBM network, as described in the preceding paragraph. We devised two alternative ways to compute the KL-divergence term of the VAE model with Bernoulli prior—a stochastic Monte Carlo estimator and an analytic form. Whereas previous studies have employed DVAEs primarily for the purpose of generation [26–28, 30, 32, 45–47], we explore and compare the performance of our Gaussian, Bernoulli, and RBM models in detecting anomalies in various datasets of aeronautical data[7]. Moreover, we introduce the tunable hyperparameter β into the ELBO objective, based on concepts described in [21, 22, 41], to optimize anomaly-detection performance. To the best of our knowledge, the studies described in this paper are the first to apply VAEs with discrete latent variables to anomaly-detection problems in the realm of aerospace. We describe the training of our models, including a detailed characterization of the RBM model's training process, and report anomaly-detection performance with nonoptimal hyperparameters, in addition to performance with optimized hyperparameters. We also investigate the ability of our VAE models to transfer to an unseen dataset and probe the robustness of the DVAE model with RBM prior to changes in anomaly type and phase of flight. We want to find out if the anomaly-detection performance of a VAE with discrete latent space is competitive with that of a VAE with Gaussian prior and continuous latent variables, the standard choice of VAE type. Also, if a classical deep generative model with discrete latent variables exhibits a performance that is comparable or superior to that of a continuous-variable counterpart, it is worth exploring if a quantum-enhanced version of the discrete model, which can exploit complex, classically not accessible, correlations brought about by quantum states, can achieve a performance that exceeds that of the fully classical discrete model.

We conducted three experimental studies for this paper. The first ('baseline') study uses a dataset with a drop-in-airspeed anomaly during takeoff. It investigates the models' training behavior and compares the anomaly-detection performance of the Gaussian, Bernoulli, and RBM models. Our second study investigates the ability of our trained models to generalize (transfer) to a new dataset containing the same anomaly. The models employed in this study operate with hyperparameters optimized for the dataset used in the baseline study. Our final study examines the RBM model's robustness to changes in anomaly type and phase of flight, by evaluating the model's performance on a new dataset with a delay-in-flap-deployment anomaly during approach to landing. For this study, the model's hyperparameters were re-tuned and the model was re-trained on the dataset used.

The structure of the paper is as follows. In section 2, we review causal generative models, that is, probabilistic models that reconstruct input data from latent variables. We also describe prior distributions used in generative modeling with continuous and discrete latent variables. Section 3 covers VAEs with continuous and discrete latent spaces. We describe the β-VAE, which regulates the trade-off between the autoencoding and KL terms of the ELBO, and introduce alternative formulations of RBM prior networks in the VAE's latent space. In section 4, we describe the methodology we used to evaluate our models' anomaly-detection performance. In section 5, we present the experimental findings of our three studies, outlined in the preceding paragraph, and discuss model design and the influence on performance of model

[7] For simplicity's sake, we frequently refer to the VAE models with Gaussian, Bernoulli, and RBM priors as the Gaussian, Bernoulli, and RBM models, respectively, in this paper; the longer, more correct, expression is used interchangeably with the abbreviated version.

hyperparameters and of the application of normalizing transformations to anomaly scores. We present our conclusions in section 6. The appendices and supplementary material contain additional information on concepts and experiments.

2. Causal generative modeling

The goal of generative modeling is to estimate the probability distribution of the input data, $p(\mathbf{x})$, which is unknown but assumed to exist. The distribution of synthetic data points, estimated by the model, is called the marginal distribution, $p_{\boldsymbol{\theta}}(\mathbf{x})$ (where $\boldsymbol{\theta}$ denotes the model parameters). The goal is to make $p_{\boldsymbol{\theta}}(\mathbf{x})$ as close to $p(\mathbf{x})$ as possible. In order to accomplish this objective, representational modeling computationally analyzes the statistical structure of a dataset and attempts to identify a set of latent (unobserved) variables $\mathbf{z}$ that represent the dominant features of the dataset. Latent variables are also known as 'causes' [5]. A graphical model of a directed generative model with latent variables is depicted in figure 1. Conceptual details on the generative and recognition models used in causal generative modeling can be found in appendix A. Generative modeling is well suited for unsupervised learning: lacking supervisory information, a model's performance is determined by the ability of its latent variables to represent and reproduce the statistical structure of the input variables [5].

2.1. Prior distributions

In a directed generative model, the model distribution $p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})$ is explicitly factored into the generative distribution $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$, the distribution of the generative model's output given a latent-space realization, and the model's prior distribution $p_{\boldsymbol{\theta}}(\mathbf{z})$. The simplest generative-model priors are factorized standard normal or Bernoulli distributions:

$$\begin{aligned} p_{\boldsymbol{\theta}}(\mathbf{z}) &= \mathcal{N}(\mathbf{z};\mathbf{0},\mathbf{I}) \equiv \prod_{l=1}^{L}\mathcal{N}(z_l;0,1), \\ p_{\boldsymbol{\theta}}(\mathbf{z}^d) &= \mathcal{B}(\mathbf{z}^d;\mathbf{0.5}) \equiv \prod_{l=1}^{L}\mathcal{B}(z_l^d;0.5), \end{aligned} \tag{1}$$

where the subscript l indexes a latent variable. Also, we use the symbolic expression z^d to denote a discrete latent variable (to distinguish it from a continuous latent variable, expressed as z). The posterior distributions factorize accordingly [29].

Boltzmann machine (BM) priors are more expressive and capable of representing complex multi-modal probability distributions [29]. Since Boltzmann priors are intractable in models with variable counts encountered in practical use, it is common to use Markov chain Monte Carlo (MCMC) sampling to estimate gradients. The efficiency of MCMC sampling is greatly increased [30] by stipulating a bipartite connectivity between the groups of visible units $\mathbf{z}_v^d$ and hidden units $\mathbf{z}_h^d$ of the BM, without lateral connections between the units within either group, i.e. an RBM [44]. The visible and hidden units correspond to the input and latent variables, respectively, of an undirected generative model and are binary (0 or 1) in the RBMs we employed in the experiments reported in this paper. An RBM prior is given by [29, 48, 49]:

$$\begin{aligned} p_{\boldsymbol{\theta}}(\mathbf{z}_v^d,\mathbf{z}_h^d) &= \mathrm{e}^{-E_{\boldsymbol{\theta}}(\mathbf{z}_v^d,\mathbf{z}_h^d)}/Z_{\boldsymbol{\theta}}, \quad Z_{\boldsymbol{\theta}} \equiv \sum_{\mathbf{z}_v^d,\mathbf{z}_h^d}\mathrm{e}^{-E_{\boldsymbol{\theta}}(\mathbf{z}_v^d,\mathbf{z}_h^d)}, \\ E_{\boldsymbol{\theta}}(\mathbf{z}_v^d,\mathbf{z}_h^d) &= -(\mathbf{z}_v^d)^{\mathrm{T}}\mathbf{W}\mathbf{z}_h^d - \mathbf{a}^{\mathrm{T}}\mathbf{z}_v^d - \mathbf{b}^{\mathrm{T}}\mathbf{z}_h^d. \end{aligned} \tag{2}$$

In (2), $E_{\boldsymbol{\theta}}(\mathbf{z}_v^d,\mathbf{z}_h^d)$ is the energy function of visible and hidden units and $Z_{\boldsymbol{\theta}}$ the normalizing constant (partition function) of prior $p_{\boldsymbol{\theta}}(\mathbf{z}_v^d,\mathbf{z}_h^d)$; $\mathbf{W}$, $\mathbf{a}$, and $\mathbf{b}$ are the weight matrix between visible and hidden units and the visible and hidden bias vectors, respectively. The conditional distributions of the hidden given the visible units and of the visible given the hidden units are then given by [6]:

$$\begin{aligned} p_{\boldsymbol{\theta}}(\mathbf{z}_h^d|\mathbf{z}_v^d) &= \prod_{l=1}^{L} p_{\boldsymbol{\theta}}((z_h^d)_l|\mathbf{z}_v^d) = \prod_{l=1}^{L}\sigma(b_l + \mathbf{W}_{\cdot l}^{\mathrm{T}}\mathbf{z}_v^d), \\ p_{\boldsymbol{\theta}}(\mathbf{z}_v^d|\mathbf{z}_h^d) &= \prod_{k=1}^{K} p_{\boldsymbol{\theta}}((z_v^d)_k|\mathbf{z}_h^d) = \prod_{k=1}^{K}\sigma(a_k + \mathbf{W}_{k\cdot}\,\mathbf{z}_h^d), \end{aligned} \tag{3}$$

where σ denotes the logistic function and $\mathbf{W}_{\cdot l}^{\mathrm{T}}$ is a vector consisting of the transpose of the lth column of weight matrix $\mathbf{W}$ and $\mathbf{W}_{k\cdot}$ a vector consisting of the kth row of $\mathbf{W}$. Because of the absence of lateral

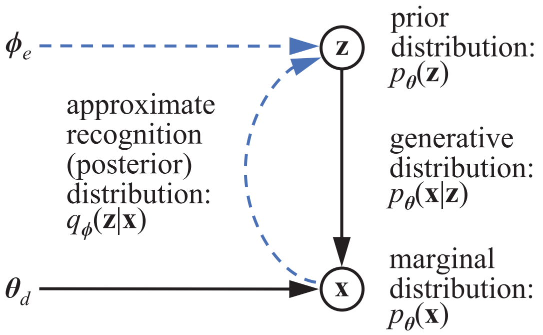

ϕ_e
$\mathbf{z}$
prior distribution: $p_\theta(\mathbf{z})$
approximate recognition (posterior) distribution: $q_\phi(\mathbf{z}|\mathbf{x})$
generative distribution: $p_\theta(\mathbf{x}|\mathbf{z})$
θ_d
$\mathbf{x}$
marginal distribution: $p_\theta(\mathbf{x})$

Figure 1. Generative models with latent variables, $\mathbf{z}$, can be represented as probabilistic graphical models that depict conditional relationships among variables. In a directed generative model, the model distribution, $p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{z})$, is explicitly factored into the generative (decoder) distribution, $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$, and the model's prior distribution over latent variables, $p_{\boldsymbol{\theta}}(\mathbf{z})$. The marginal distribution, $p_{\boldsymbol{\theta}}(\mathbf{x})$, is obtained by marginalizing (integrating or summing) over the latent variables. Since the computation of the true posterior, $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$, is intractable, variational inference substitutes it with an approximating posterior (encoder), $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$. Black, solid arrows denote the generative model and blue, dashed arrows the variational approximation to the intractable posterior. In the VAE models used to conduct the experiments described in this paper, the posterior and prior over latent variables are either continuous Gaussian distributions [posterior: $\mathcal{N}(\mathbf{z};\boldsymbol{\mu},\mathrm{diag}(\boldsymbol{\sigma}^2))$, prior: $\mathcal{N}(\mathbf{z};\mathbf{0},\mathbf{I})$] or discrete Bernoulli or Boltzmann distributions [posterior: $\mathcal{B}(\mathbf{z}^d;\mathbf{q})$, prior: $\mathcal{B}(\mathbf{z}^d;\mathbf{0.5})$ or $\mathrm{e}^{-E_{\boldsymbol{\theta}}(\mathbf{z}_v^d,\mathbf{z}_h^d)}/Z_{\boldsymbol{\theta}}$]. We use the symbolic expression z^d to denote a *discrete* latent variable.

connections between visible units and between hidden units, the conditional probabilities in (3) can be determined in one fell swoop, using block Gibbs sampling. In block Gibbs sampling, the values of all hidden units are updated at once by sampling from their conditional distribution given the visible units [top equation of (3)]. Then, the values of the visible units are updated analogously [bottom equation of (3)]. This process can be repeated for an arbitrary number of iterations. When Gibbs sampling is performed for an infinite number of steps, it is guaranteed to converge to the stationary distribution $p_{\boldsymbol{\theta}}(\mathbf{z}_v^d,\mathbf{z}_h^d)$ of the RBM model [50, 51], and computationally efficient techniques to learn the model distribution have been developed [52–54].

3. Variational autoencoders

VAEs are directed generative models with latent variables that approximate the intractable true posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ via variational inference and maximize an ELBO objective $\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x})$ (see figure 1 and appendix B). The ELBO can be re-written as [10]:

$$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] - D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z})). \tag{4}$$

The first term of (4) is the autoencoding term. Maximizing it maximizes the fidelity of reconstruction because the greater the autoencoding term, the greater is the similarity between the data distribution $p(\mathbf{x})$ and the generative distribution $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ when $\mathbf{z}$ is sampled from the approximate posterior distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ of the encoder. Conversely, the second term is maximized by minimizing the KL divergence between the approximate posterior and prior, which corresponds to minimizing the mutual information between $\mathbf{x}$ and $\mathbf{z}$. Consequently, the autoencoding term attempts to maximize the mutual information between data and latents and the KL term seeks to minimize it [29]. Eventually, the latent-space's information content will depend on the trade-off between the two terms, which, in turn, is determined by the flexibility and expressiveness of the variational approximation $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$, the structure of the generative model, and the training method [29, 31].

Given training set $\mathcal{D} = \{\mathbf{x}^{(n)}\}_{n=1}^{N}$ consisting of N i.i.d. samples from $p(\mathbf{x})$, the ELBO for a minibatch of training data is given as the average of the ELBOs of the minibatch instances [10]:

$$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathcal{M}^{(i)}) = \frac{1}{M}\sum_{\mathbf{x}\in\mathcal{M}^{(i)}} \mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}), \tag{5}$$

where the minibatch $\mathcal{M}^{(i)} = \{\mathbf{x}^{(i,m)}\}_{m=1}^{M}$ contains M data points randomly sampled from $\mathcal{D}$ with N data points.

3.1. VAE with factorized Gaussian prior

The ELBO objective given in (4) contains expectations of functions of the latent variables $\mathbf{z}$ with regard to the variational posterior $q_\phi(\mathbf{z}|\mathbf{x})$, which can be written as $\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[f(\mathbf{z})]$, where f denotes an arbitrary function.

To train the model with minibatch stochastic gradient descent starting from random initializations of the model parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, we need to calculate gradients of such expectations [29, 55]. Procedures to obtain unbiased gradients with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are described in appendix C.

We estimated the Gaussian parameters $\boldsymbol{\mu}$ and $\log \boldsymbol{\sigma}^2$ by means of linear layers at the end of the VAE's encoder (see appendix G). The $\log \boldsymbol{\sigma}^2$ estimate was then routed through a softplus activation for numerical stability. As a result of applying rectified-linear-unit (ReLU) activations to each layer of the encoder and decoder, the mean and log variance of the approximate posterior, $\boldsymbol{\mu}$ and $\log \boldsymbol{\sigma}^2$, are nonlinear functions of the input data $\mathbf{x}$ and the variational parameters $\boldsymbol{\phi}$ [10].

Moreover, when the VAE prior is given by a factorized Gaussian, the KL-divergence term in the ELBO objective [(4)] can be expressed in closed form. The ELBO is then estimated as:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) \simeq \frac{1}{S}\sum_{s=1}^{S} \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}^{(s)}) + \frac{1}{2}\sum_{l=1}^{L}(1 + \log \sigma_l^2 - \mu_l^2 - \sigma_l^2), \tag{6}$$

where $\mathbf{z}^{(s)} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}^{(s)}$, $\boldsymbol{\epsilon}^{(s)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and l indexes a latent variable [10].

3.2. β-VAE

Higgins *et al* [21] modified the VAE objective to reduce the entanglement between latent variables. Each latent variable z_l is to represent a meaningful domain-specific attribute that varies along a continuum when z_l is varied. In order to promote this disentangling property in the latent variables $\mathbf{z} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$, the authors introduce a constraint over the posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ from which they are derived, making it more similar to a prior $p_{\boldsymbol{\theta}}(\mathbf{z})$. This condition restrains latent-space capacity and stimulates statistical independence between individual latent-space variables. The ELBO objective [(4)] of a β-VAE is given as (see appendix D for derivation):

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}, \beta) = \mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] - \beta D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z})). \tag{7}$$

The parameter β is used to balance the trade-off between the fit of the reconstructed data, $\hat{\mathbf{x}}$, to the input data, $\mathbf{x}$, imposed by the autoencoding (reconstruction) term (low β), and the fit of the posterior, $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$, to the prior, $p_{\boldsymbol{\theta}}(\mathbf{z})$, via the KL term (high β). If $\beta = 0$, the β-VAE model is identical to an autoencoder, and if $\beta = 1$, the model corresponds to a regular VAE. We would like to note that we do not use β to explicitly disentangle the latent space but employ it as a regularization hyperparameter that requires tuning, an approach pioneered in [41].

3.3. VAE with discrete latent space

Several approaches have been developed to circumvent the non-differentiability problem affecting models with discrete latent units [56–59]. In VAE models, the reparameterization trick has been extended by either the incorporation of smoothing functions [26] or the relaxation of discrete latent variables into continuous ones [24, 30, 60]. In this work, we employ the Gumbel-softmax trick, which relaxes a discrete categorical distribution into a continuous concrete (or Gumbel-softmax) distribution [24, 60]. Appendix E describes the reparameterization trick and the obtention of unbiased gradients with respect to the variational parameters of a VAE with a discrete latent space. We estimated the log odds of the relaxed Bernoulli approximate posterior probabilities, $\log \boldsymbol{\alpha}^{\mathbf{q}}$, by means of a linear layer at the end of the VAE's encoder (see appendix G).

Discrete VAEs can be implemented with Bernoulli [bottom equation in (1)] or RBM [(2)] priors. The KL term of the ELBO of a VAE with a Bernoulli prior can be expressed as:

$$\begin{aligned} D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}^d|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}^d)) &= \mathbb{E}_{\mathbf{z}^d\sim q_{\boldsymbol{\phi}}(\mathbf{z}^d|\mathbf{x})}\left[\log \frac{q_{\boldsymbol{\phi}}(\mathbf{z}^d|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}^d)}\right] \\ &= \mathbb{E}_{\mathbf{z}^d\sim \mathcal{B}(\mathbf{q})}\left[\sum_{l=1}^{L}(z_l^d \log q_l + (1 - z_l^d)\log(1 - q_l) \right. \\ &\qquad \left. - z_l^d \log 0.5 - (1 - z_l^d)\log(1 - 0.5))\right], \end{aligned} \tag{8}$$

where q_l stands for the parameter of the latent Bernoulli variable $z_l^d \sim \mathcal{B}(q_l)$ and 0.5 is the parameter of the Bernoulli prior distribution, $\mathcal{B}(p_l = 0.5)$. A more detailed derivation of the above expression and implementation details as well as an analytic expression for the KL term in the ELBO objective [(4)] of a VAE with Bernoulli prior are given in appendix F.

Unbiased gradients of the KL term of the ELBO of a VAE with an RBM prior with respect to the generative and variational parameters can be obtained as:

$$\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z})) = \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}}\{\mathbb{E}_{\boldsymbol{\rho}\sim\mathcal{U}(\mathbf{0},\mathbf{1})}[\log q_{\boldsymbol{\phi}}(\mathbf{z}(\boldsymbol{\phi},\boldsymbol{\rho})|\mathbf{x})] + \mathbb{E}_{\boldsymbol{\rho}\sim\mathcal{U}(\mathbf{0},\mathbf{1})}[E_{\boldsymbol{\theta}}(\mathbf{z}(\boldsymbol{\phi},\boldsymbol{\rho}))] - \mathbb{E}_{\tilde{\mathbf{z}}^d\sim p_{\boldsymbol{\theta}}(\tilde{\mathbf{z}}^d)}[E_{\boldsymbol{\theta}}(\tilde{\mathbf{z}}^d)]\}, \tag{9}$$

where the gradients of the log prior probability are given, as usual, as the difference between a positive and negative phase. The symbolic expression $\tilde{\mathbf{z}}^d$ denotes 'fantasy states,' i.e. values of the latent variables produced by the RBM model (prior) distribution, which remain discrete and are not relaxed during training [30, 31]. In the expression above, we have highlighted the fact that the positive-phase energy $[E_{\boldsymbol{\theta}}(\mathbf{z}(\boldsymbol{\phi},\boldsymbol{\rho}))]$ and the negative-phase energy $[E_{\boldsymbol{\theta}}(\tilde{\mathbf{z}}^d)]$ are calculated [according to the bottom equation of (2)] using relaxed posterior samples ($\mathbf{z}$) and discrete model samples (fantasy states $\tilde{\mathbf{z}}^d$), respectively. The training objective is to make the model distribution, $p_{\boldsymbol{\theta}}(\tilde{\mathbf{z}}^d)$, as similar as possible to the posterior distribution, $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$. See appendix F for details on the KL-divergence term in the ELBO of a DVAE/RBM model and its gradients. We used the persistent-contrastive-divergence (PCD) algorithm [53] to evolve the conditional distributions of the 'visible' and 'hidden' layers of fantasy states of the negative phase by means of Gibbs sampling, starting from initialization to zero. In PCD, the chains of the fantasy states' values are persistent and continue to evolve over cycles of training (minibatches), without re-initialization at the beginning of the cycle. The PCD algorithm is characterized by short mixing times (fast convergence to the stationary distribution) because the weight updates repel the persistent chains from their current states by raising the energies of the states [50]. It should be noted that this form of training does not require knowledge of the (intractable) partition function $Z_{\boldsymbol{\theta}}$. Hence, our VAE model with RBM prior is an energy-based model whose training is based on (unnormalized) prior energies rather than (normalized) prior probabilities.

3.4. Latent Boltzmann networks

In a VAE with an RBM prior, the RBM network is located in the latent space; there are no visible RBM units corresponding to input data as in a standalone RBM. Also, by necessity, the latent variables $\mathbf{z}$ of the positive phase are continuous (because they are sampled from the approximate posterior distribution, which is relaxed via the Gumbel-softmax procedure described above and in appendix E during training in order to make the ELBO differentiable). On the other hand, the RBM model samples (fantasy states) $\tilde{\mathbf{z}}^d$ remain discrete variables, as indicated by the superscript, and are not relaxed during training.

Rolfe [26] first developed a DVAE with RBM prior. The model applied the spike-and-exponential transformation to the posterior latents to make them differentiable. Khoshaman and Amin [30] then modified the DVAE/RBM model by using the Gumbel-softmax trick to bring about the continuous relaxation of the DVAE's latent variables; the authors termed their DVAE model with RBM prior and Gumbel-softmax relaxation 'GumBolt.' Vinci *et al* [31] introduced a quantum version of the GumBolt model, based on Amin *et al* [61] and Khoshaman *et al* [29]. These authors split up the posterior latent units $\mathbf{z}$ into two portions of equal size (denoted as $\mathbf{z}_l$ and $\mathbf{z}_r$ in figure 2) to implement between them the positive phase of the RBM model according to (9) and the energy function given in (2). A corresponding approach is taken for the fantasy states $\tilde{\mathbf{z}}^d$ of the negative phase. We designate an RBM model with such variables a 'bipartite latent-space RBM.' In such an RBM model, there is no difference in kind between the 'visible' and 'hidden' units (for example, $\mathbf{z}_l$ and $\mathbf{z}_r$, respectively, in the positive phase). The fantasy states are evolved via PCD Gibbs sampling whereas the posterior latent states remain 'clamped' to their original values (VAE encoder output) and are not subjected to Gibbs sampling.

In the studies conducted for this paper, we have adopted a slightly different approach. We consider the posterior latent variables $\mathbf{z}$ to be inputs of the positive phase of the latent-space RBM ($\mathbf{z}_v$) and add an equal number of hidden units ($\mathbf{z}_h^d$) to the model. The values of the hidden units of the positive phase are determined by one-step Gibbs sampling, given the values of the visible units [see top equation of (3) for the hidden units' distribution][8]. The values of the 'visible' and 'hidden' units of the negative phase ($\tilde{\mathbf{z}}_v^d$ and $\tilde{\mathbf{z}}_h^d$, respectively) are evolved by PCD Gibbs sampling, as in the bipartite latent-space model. We call a model with

[8] During training, the visible units of the positive phase $\mathbf{z}$, which correspond to the VAE's latent variables, are continuous because they are Bernoulli variables relaxed via the Gumbel-softmax trick to make the objective function differentiable. The fantasy states of the negative phase $\tilde{\mathbf{z}}^d$, by contrast, remain discrete variables and are not relaxed during training. However, the variable type (continuous or discrete) of the hidden units of the positive phase is not obvious. The values of the positive phase's visible and hidden units are used to compute the positive phase's energy $E_{\boldsymbol{\theta}}(\mathbf{z}_v, \mathbf{z}_h^{(d)})$, which does not depend on the variational parameters $\boldsymbol{\phi}$ [see bottom equation of (2)]. We evaluated all applicable combinations of continuous and discrete units in the formula for the energy of the positive phase (continuous visible and continuous hidden, continuous visible and discrete hidden, and discrete visible and discrete hidden). The combination of continuous visible units and discrete hidden units produced the best performance, and we chose this combination, based on this empirical observation, as indicated in figure 2. We will address this question more rigorously in future research.

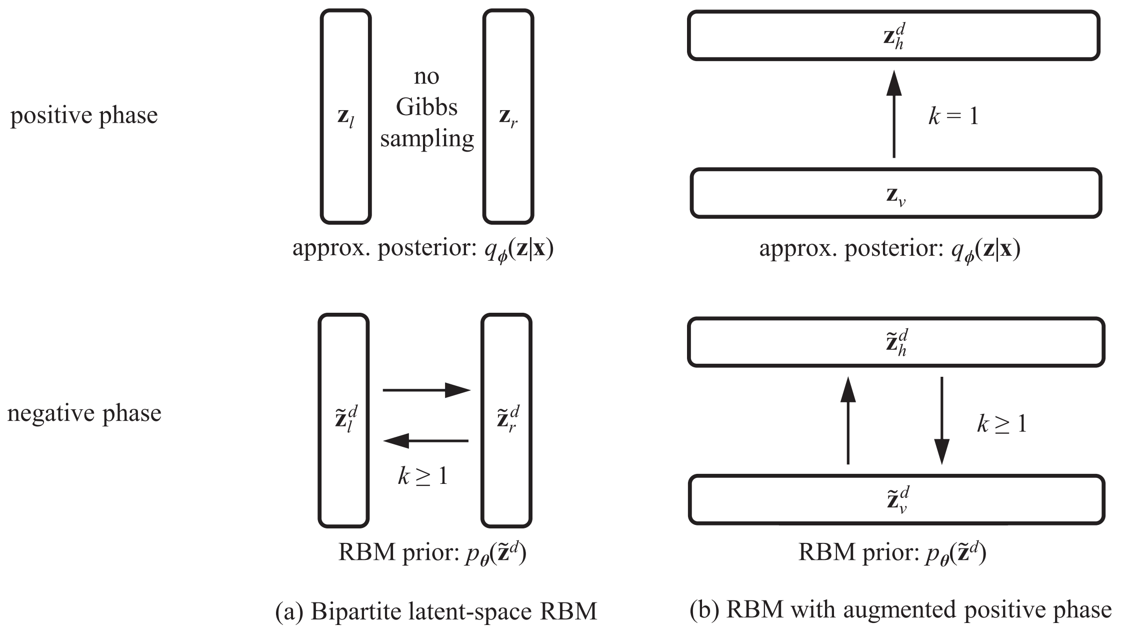

positive phase
no Gibbs sampling
$\mathbf{z}_l$
$\mathbf{z}_r$
approx. posterior: $q_\phi(\mathbf{z}|\mathbf{x})$
$\mathbf{z}_h^d$
$k = 1$
$\mathbf{z}_v$
approx. posterior: $q_\phi(\mathbf{z}|\mathbf{x})$
negative phase
$\tilde{\mathbf{z}}_l^d$
$\tilde{\mathbf{z}}_r^d$
$k \geq 1$
RBM prior: $p_\theta(\tilde{\mathbf{z}}^d)$
$\tilde{\mathbf{z}}_h^d$
$k \geq 1$
$\tilde{\mathbf{z}}_v^d$
RBM prior: $p_\theta(\tilde{\mathbf{z}}^d)$
(a) Bipartite latent-space RBM
(b) RBM with augmented positive phase

Figure 2. Contrasting architectures of RBM prior models. (a) Prior model in which an RBM is implemented between two subdivisions of the VAE's latent space, obtained by dividing the latent units $\mathbf{z}$ and $\tilde{\mathbf{z}}^d$, sampled from, respectively, the approximate posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ and model prior $p_{\boldsymbol{\theta}}(\tilde{\mathbf{z}}^d)$, into two equally sized parts each (here termed 'bipartite latent-space RBM') [26, 29–31]. (b) RBM prior model in which a true hidden layer is added to the layer of posterior latents, which constitute the 'visible' layer of the RBM's positive phase (here termed 'RBM with augmented positive phase'). Here, the symbolic expressions for the approximate posterior and prior distributions [$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ and $p_{\boldsymbol{\theta}}(\tilde{\mathbf{z}}^d)$, respectively] highlight the fact that the posterior latents are relaxed into continuous random variables $\mathbf{z}$ during training whereas the prior latents (fantasy states) remain discrete variables $\tilde{\mathbf{z}}^d$. In a bipartite latent-space RBM, the latent states $\mathbf{z}$ of the approximate posterior remain 'clamped' to their original values (VAE encoder output) whereas the fantasy states $\tilde{\mathbf{z}}^d$ of the negative phase are evolved via PCD Gibbs sampling. In this work, we treat the latent variables $\mathbf{z}$ of the approximate posterior as the visible units of the RBM ($\mathbf{z}_v$) and add an equal number of hidden units ($\mathbf{z}_h^d$), thus augmenting the RBM's positive phase. The values of $\mathbf{z}_h^d$ are determined by one-step Gibbs sampling, given the values of $\mathbf{z}_v$. The values of the 'visible' and 'hidden' units of the negative phase are evolved by PCD Gibbs sampling, as in the bipartite latent-space model. The symbol k denotes the number of Gibbs-sampling steps.

such features an 'RBM with augmented positive phase.' The contrasting architectures of the two models are shown in figure 2. We chose the augmented model rather than the bipartite latent-space one to conduct our experiments because it had demonstrated a somewhat more dynamic training and a slightly better performance in preliminary experiments that did not comprehensively evaluate and compare the training behavior and performance characteristics of the two RBM-model versions.

4. Evaluation of anomaly-detection performance

For the experiments reported in this paper, we trained 16 models (10 for the experiments assessing performance on the baseline dataset with nonoptimal hyperparameters; section 5.1.2) independently, and we report mean $+/-$ standard deviation. The reconstruction error of the ELBO [negative of left term in (6)] can be operationalized by the mean squared error (MSE) between the training data and the decoder output (reconstructed training data). We used this error metric when the input data were normalized by mean centering and scaling to unit variance (z scores), which was the case for the datasets with drop-in-airspeed anomaly during takeoff. The MSE between the original and reconstructed training data is:

$$\mathrm{MSE}_{\mathbf{x}} = \frac{1}{S}\sum_{s=1}^{S} ||\mathbf{x}^{(s)} - \hat{\mathbf{x}}^{(s)}||_2^2, \tag{10}$$

where $\mathbf{x}$ symbolizes the input data and $\hat{\mathbf{x}}$ their reconstructions and the sum is taken over a minibatch of training data.

We used a different error metric to estimate the reconstruction error when the training data were normalized to lie between zero and one using the transformation

$$\boldsymbol{\xi}' = \frac{\boldsymbol{\xi} - \min\{\boldsymbol{\xi}\}}{\max\{\boldsymbol{\xi}\} - \min\{\boldsymbol{\xi}\}}, \tag{11}$$

where $\boldsymbol{\xi}'$ symbolizes the transformed data. This was the case for the dataset with a delay-in-flap-deployment anomaly during approach to landing. In this case, we used the binary cross entropy (BCE) to estimate the reconstruction error:

$$\mathrm{BCE}_{\mathbf{x}} = -\frac{1}{S}\left(\sum_{s=1}^{S}\sum_{j=1}^{J} x_j^{(s)} \log \hat{x}_j^{(s)} + (1 - x_j^{(s)}) \log(1 - \hat{x}_j^{(s)})\right), \tag{12}$$

where the sums are over input instances s and features j. Our experiments had shown that the BCE error metric captured the reconstruction error more accurately when the input data were scaled to the interval [0, 1].

Since nominal data points are much more prevalent than anomalous ones, a generative model primarily learns patterns exhibited by nominal data, and, therefore, their reconstruction errors (MSE or BCE) tend to be smaller than the errors of anomalous points. However, a powerful encoder-decoder model without regularization will also fit anomalous data points, which is undesirable when the reconstruction error is used as the metric to identify anomalies, as is the case in our studies. To discourage the fitting to anomalous instances, our models are *variational* (rather than pure) autoencoders. The regularizing KL-divergence term in the ELBO of a VAE [(4)] penalizes the divergence between the approximate posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ and prior $p_{\boldsymbol{\theta}}(\mathbf{z})$. To control the extent of fitting to the training data (via the autoencoding term) relative to the strength of regularization (via the KL term), we introduce the hyperparameter β into the objective function [see (7)], based on the method developed in [21]. β regulates the relative weighting of the autoencoding and KL terms of the ELBO. When β is chosen properly, in such a way that the reduction in KL divergence due to the similarity between posterior and prior outweighs the rise in reconstruction error due to the lack of fit to (the few) anomalous data points, the model can be induced to preferentially fit data in the nominal majority class. Hence, optimal anomaly detection depends on the careful tuning of the hyperparameter β.

Once a VAE model is trained with empirically optimized hyperparameters, we use the reconstruction errors per training instance, e, to determine an anomaly-score threshold. The threshold is based on the assumption that the data (nominal and anomalous instances) are normally distributed and is specified as:

$$thr = \langle e \rangle + z\Delta e, \tag{13}$$

where the angle brackets and Δ denote, respectively, the mean and standard deviation over training instances, and z the z score, derived from the (known) percentage of anomalies in the training set using the quantile function. Once the threshold is determined based on the training data, we identify anomalies in the test data by calculating the anomaly score (reconstruction error) for each test-data instance and comparing it to the above threshold; instances with an anomaly score below the threshold are classified as nominal, and instances with a score above the threshold are considered anomalous. This two-step process, of computing the anomaly-score threshold based on the training data and calculating anomaly scores from test data, ensures that the proportion of anomalies in the test set needs not to be the same as in the training set: the threshold is simply determined from the training set, and, independent thereof, test-set instances with anomaly scores less than the threshold are considered nominal and test instances with scores greater than it are considered anomalous. In practical situations, the fractions of anomalies in the training and test sets might often be similar (as in our experiments), but this will not always be the case, and our models and anomaly-detection methodology can accommodate such cases.

Anomaly scores given by the BCE error metric are reasonably normally distributed. However, anomaly scores corresponding to the MSE metric are considerably skewed to the right and possess a long right tail. We applied various normalizing transformations to such anomaly scores, including the square-root, natural-logarithm, and inverse (reciprocal) transformation. On average, the logarithm produced the best anomaly-detection performance (on the validation set). For this reason, we applied a log transformation to the reconstruction errors of training-set instances and the anomaly scores of test-set instances derived from datasets normalized with standard (z-score) scaling. We sampled both the (log-transformed) training-set reconstruction errors and the (log-transformed) anomaly scores of the test set ten times each per data point. We then used the average (log-transformed) reconstruction errors per training instance e to compute the anomaly-score threshold thr according to (13) and the average (log-transformed) anomaly scores of the test set to classify data as nominal or anomalous. The thus predicted data labels, determined in an unsupervised way, were then compared with the known true data labels to compute performance metrics.

In all studies, we assume that the nominal data are the negative class and that the anomalous data are the positive class. We assess model performance with three metrics—precision, recall, and F1 score—specified as:

$$\begin{aligned} \text{precision} &= \frac{TP}{TP+FP}, \\ \text{recall} &= \frac{TP}{TP+FN}, \\ \text{F1 score} &= \frac{2}{\text{recall}^{-1}+\text{precision}^{-1}}, \end{aligned} \tag{14}$$

where TP represents true positives: correctly identified anomalies, FP false positives (alarms): nominal instances incorrectly categorized as anomalous, and FN false negatives: missed anomalies or instances that were incorrectly classified as nominal.

5. Experimental results and discussion

All VAE models were implemented in Python using the PyTorch deep-learning library [62]. Our VAE model with Gaussian prior is an upgraded version of the CVAE model introduced in [41], which achieved state-of-the-art performance in detecting anomalies in aviation time series. In all models, the Adam optimizer was used, with a learning rate of 3×10^{-4} and default momentum parameters [63]. We used minibatch-based optimization. Minibatches of mutually exclusive training-set instances were re-shuffled for each epoch of training. All minibatches comprised 128 training-set instances, except for the minibatches used for post-training in the transferability study (section 5.2), which contained 32 instances. We used validation sets to determine combinations of hyperparameter values with good performance. Except for the final layer of the encoder, which outputs the estimated parameters of the approximate posterior and the reparameterized latent variables (and effects the relaxation of discrete latent variables in discrete-variable models), all models use the same encoder and decoder architecture presented in appendix G.

5.1. Baseline study: drop in airspeed during takeoff

We determined the baseline performance of the VAE models with Gaussian, Bernoulli, and RBM priors on a dataset of departing flights with drop-in-airspeed anomaly. Subject matter experts ascertained that if the speed of an aircraft drops by more than 20 knots during the first minute of flight, an adverse event might ensue, and, therefore, data points with such a property are classified as anomalous. The dataset contains 27 346 instances of flight-operation data from commercial flights, with 657 (2.40%) anomalies. It comprises time series of primarily 1 Hz recordings of seven flight parameters, including position (i.e. altitude), orientation (i.e. angle of attack and pitch angle), speed (i.e. computed airspeed and wind speed), and binary state variables describing whether the auto-throttle and lateral autopilot modes were on or off. The data were acquired in real time aboard the aircraft and downloaded by the airline once the aircraft had reached the destination gate. The time series of each data instance span the first 60 s of the initial ascent after becoming airborne. The drop-in-airspeed anomaly is not necessarily the only type of anomaly in the dataset, and the true number of operationally significant anomalies is unknown.

5.1.1. Model training

For this baseline study, we randomly divided the data into training (60%), validation (20%), and test (20%) sets. We used the training set to train the models, the validation set to monitor overfitting and select hyperparameters, and the test set to assess model performance in an unbiased way. Since the input data were normalized by mean centering and scaling to unit variance, the MSE [(10)] was used to estimate the reconstruction error. Models were trained for 400 epochs. Model weights tended to converge at about 100 epochs. We did not observe any overfitting with increasing training time, as visualized by the change over time of the model's loss [negative of the β-ELBO objective (7), sampled over minibatches] when evaluated on the validation set (figure S1 in the supplementary material).

We assessed many combinations of hyperparameters, separately for each model type (Gaussian, Bernoulli, RBM), and selected the hyperparameter values that produced the best overall performance in precision, recall, and F1 score. Our approach to use anomaly-detection performance on a validation set, assessed by comparing the data labels predicted by a model with the validation set's known true labels, to tune model hyperparameters is comparable to the hyperparameter-optimization strategies employed in [41, 64]. The hyperparameters chosen for each model are given in table 1. The RBM model, which has a more flexible and expressive prior than the models with standard normal or Bernoulli prior, performed optimally at a lower latent-space dimensionality than the Gaussian and Bernoulli models. We also explored the application of a loss penalty to the RBM coupling weights $\mathbf{W}$ and the use of a sampling replay buffer, in which

Table 1. Hyperparameters used for the Gaussian, Bernoulli, and RBM models.

Model prior	No. latents[a]	beta[b]	lambda[c]	No. fant. part.[d]	Len. pers. chains[e]
Drop-in-airspeed anomaly during takeoff					
Gaussian	256 [16–512]	60 [1–100]	N/A	N/A	N/A
Bernoulli	128 [16–512]	60 [1–100]	0.1 [0.05–0.3]	N/A	N/A
RBM	64 [16–512]	60 [1–100]	0.1 [0.05–0.3]	500 [100–2000]	20 [1–100]
Delay-in-flap-deployment anomaly during approach to landing					
RBM	32 [16–512]	30 [1–100]	0.1[f]	500[g]	25 [1–100]

Numbers in brackets give the minimum and maximum hyperparameter values investigated.

[a] Number of latent units.

[b] Hyperparameter β controlling the balance between the autoencoding and KL terms.

[c] Temperature of the relaxed Bernoulli distribution.

[d] Number of fantasy particles/persistent chains.

[e] Length of persistent chains.

[f] We did not re-tune the temperature parameter for the study with delay-in-flap-deployment anomaly because in prior studies values of λ less than 0.1 had led to numerical instability in gradient computation due to insufficient smoothness during some training runs and greater values did not improve performance while increasing estimation bias.

[g] We did not re-tune this parameter for this study because performance is quite insensitive to the number of persistent chains. The insensitivity to this variable is due to the fact that negative-phase energies are averaged (over the number of chains) at the end of each sequence of Gibbs updates and only the average energies are used to calculate (unnormalized) log prior probabilities.

Table 2. Training times (400 epochs) of VAE models with Gaussian, Bernoulli, and RBM priors when trained on the combined training/validation set.

Model	Training time
Gaussian	1 h 00 min 29 s (± 56 s)
Bernoulli	58 min 27 s (± 19 s)
RBM ($k = 1$)	1 h 04 min 13 s (± 30 s)
RBM ($k = 20$)	1 h 34 min 17 s (± 19 s)

Times in parentheses indicate standard deviations.

The symbol k stands for the number of Gibbs-sampling steps.

5% randomly chosen fantasy states are not determined by the persistent chains but randomly set to 0 or 1 with equal probability [65]. In addition, we looked into KL-term annealing (‘warm-up’) [66], a hierarchical (conditional) posterior [31, 32], multiple sampling and averaging of the ELBO and its gradients [31], and the continuous Bernoulli to normalize the ELBO’s reconstruction term [67]. In the end, we did not apply any of these modifications because none of them improved the performance of our models, where applicable.

To assess model performance, the training and validation sets were combined, and the models were re-trained on the combined training/validation set. We then evaluated the performance of the models on the test set. The times of training the Gaussian, Bernoulli, and RBM models on the combined training/validation set for 400 epochs on a Skylake GPU-enhanced node of the Pleiades supercomputer[9] at the NASA Ames Research Center are shown in table 2. The Gaussian and Bernoulli models as well as the RBM model with one PCD Gibbs-sampling update during the negative phase require about the same average training time [Gaussian: 1 h 0 min 29 s, Bernoulli: 58 min 27 s, RBM ($k = 1$): 1 h 4 min 13 s]. On the other hand, the RBM model with 20 Gibbs updates requires, on average, more time to train [RBM ($k = 20$): 1 h 34 min 17 s].

Figure 3 shows the values of the latent units of the positive phase [$\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})$] and of the negative phase [$\tilde{\mathbf{z}}^d \sim p_{\boldsymbol{\theta}}(\tilde{\mathbf{z}}^d)$] averaged over latent dimensions and minibatch instances during a typical training run as well as the corresponding mean energies according to the bottom equation of (2). Values for each minibatch (171 per epoch) are shown. The figure illustrates that the mean negative-phase values (of the latent variables and energy) closely follow their positive-phase counterparts. However, the mean negative-phase values fluctuate less and are more centered. These findings indicate that the (free-running) negative phase re-produces a smoothed and partially averaged version of the structure of the VAE latent units of the (clamped) positive phase.

Training of the RBM biases and weights was dynamic, suggesting that the PCD algorithm explored well the energy landscape of the configurations of the system given by the dataset and model (figures S2 and S3 in

[9] www.nas.nasa.gov/hecc/resources/pleiades.html.

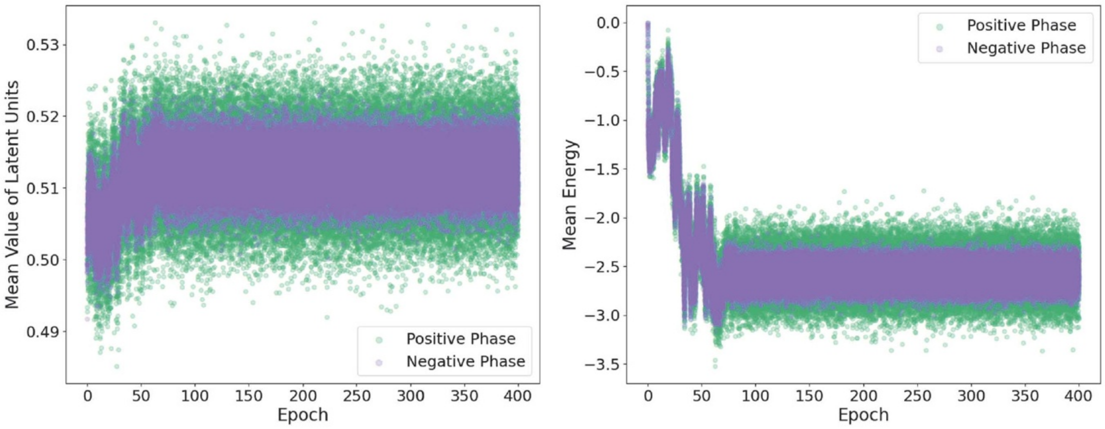
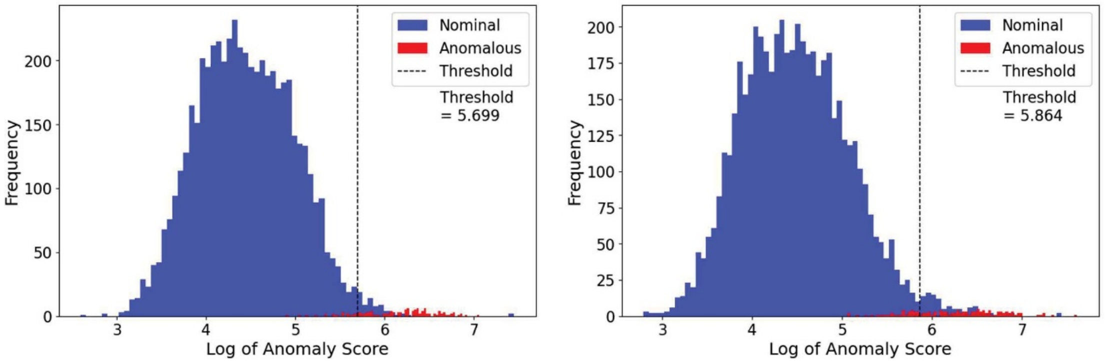


Figure 3. Values of latent units and energy, averaged over latent units and minibatch instances, of the positive phase [$\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})$] and of the negative phase [$\tilde{\mathbf{z}}^d \sim p_{\theta}(\tilde{\mathbf{z}}^d)$]. The average negative-phase values of these quantities are more centered than the corresponding average values of the positive phase, indicating that the negative phase re-produces a smoothed version of the posterior latents.



(a) Mode with good separation between nominal and anomalous instances

(b) Mode with poor separation between nominal and anomalous instances

Figure 4. Histograms of average log-transformed anomaly scores for two training modes of the RBM model. Data points to the right of the dashed threshold line are categorized as anomalies. The mode with a threshold of ∼5.70 demonstrates a good separation between nominal and anomalous data, whereas the mode with a threshold of ∼5.86 exhibits a less good separation, with a relatively high number of nominal data classified as anomalies (false positives).

the supplementary material). Histograms of average log-transformed anomaly scores of the RBM model are shown in figure 4 for two training modes. We sampled both reconstruction errors based on the training data, to determine the anomaly-score threshold, as well as the test data's anomaly scores ten times and used the sample statistics to gauge model performance, as described in section 4. We observed that models enter different modes during training and differ in their anomaly-detection performance depending on the adopted mode. The mode with a threshold of about 5.7 for log-transformed anomaly scores produced the best model performance, with F1 scores >0.65. Training with modes with a threshold $\gtrsim 5.8$, on the other hand, resulted in inferior model performance (F1 score <0.65). The superior performance of the mode with $thr \approx 5.7$ is illustrated by the cleaner separation between nominal and anomalous data. Modes with $thr \gtrsim 5.8$, on the other hand, are characterized by a greater number of false positives (nominal data to the right of the anomaly-score threshold). Other modes, with thresholds between 5.7 and 5.8, were also observed but are less common.

5.1.2. Model performance

The performance of the VAE models with Gaussian, Bernoulli, and RBM priors in the baseline study is shown in figure 5. The RBM model achieved a mean precision of 0.563, a mean recall of 0.817, and a mean F1 score of 0.666. The Gaussian model achieved a similar performance (pr 0.579, rc 0.778, f1 0.663). A notable observation is that the Bernoulli model lags behind both the RBM and the Gaussian model (pr 0.425, rc 0.596, f1 0.495). Our models demonstrate an excellent performance considering that the training was unsupervised and that the similar unsupervised CVAE model with Gaussian prior developed by [41], which

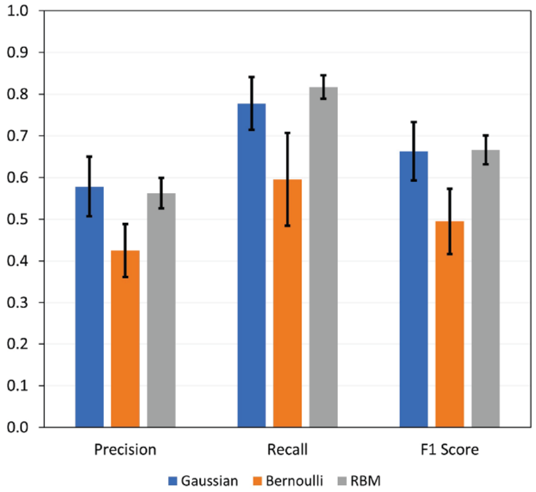
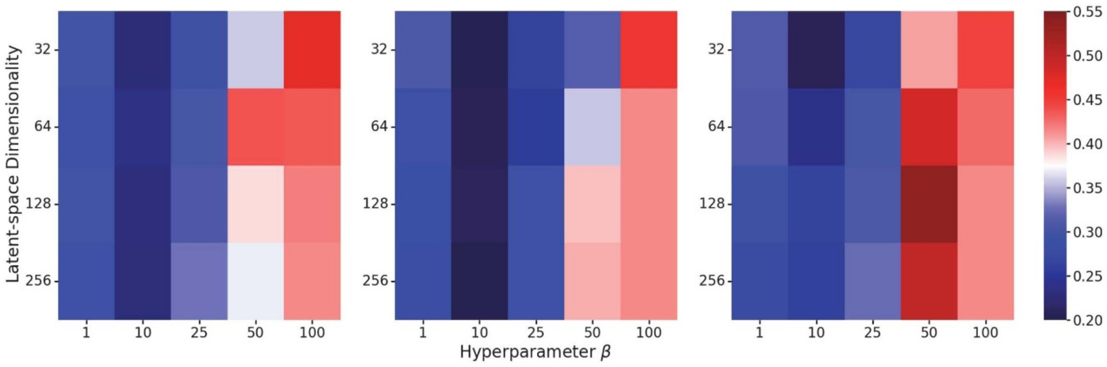


Figure 5. Performance of VAE models with Gaussian, Bernoulli, and RBM priors in the baseline study. Error bars indicate standard deviations of performance metrics among 16 independently trained models. Overall, the models demonstrate excellent performance on this unsupervised anomaly-detection task (average F1 score of 0.608). The more expressive discrete RBM model performs better than the simple discrete Bernoulli model and at a level comparable to the continuous Gaussian model. The DVAE appears to benefit from the flexibility and adaptability of the energy-based RBM prior.

achieved state-of-the-art anomaly-detection performance on aviation datasets, achieved a precision of 0.31 and recall of 0.53 on a dataset similar to the one used in our baseline study[10]. Both the Gaussian and Bernoulli models use the simplest factorized priors—the Gaussian model the continuous standard normal distribution and the Bernoulli model the discrete 'standard' Bernoulli distribution, with a parameter of 0.5. The models' similar training times also attest to their comparability (see table 2). Comparing the two models with simple factorized priors, the continuous (Gaussian) model performs better on this dataset, which is dominated by continuous time-series inputs (5 continuous time series, 2 discrete/binary ones). On the other hand, the performance deficit of the simple Bernoulli model is offset by the more expressive RBM model, both of which are discrete VAE models. Precisely the RBM model's flexibility and capability to adapt to the posterior distribution push it to a level of performance on par with the continuous Gaussian model. This observation highlights the performance boost accorded by energy-based modeling and MCMC sampling of the persistent states of the negative phase.

While training and evaluation with optimized hyperparameters allows a fair comparison between models, the approach leads to an overestimation of model performance when no validation set with labeled instances is available, as is frequently the case in practice, including in aeronautics applications. To give an impression of the performance of our models when nonoptimal hyperparameters are used, we trained models with all combinations between four different values for the latent-space dimensionality and five different settings for the hyperparameter β. The performance of models with 32, 64, 128, and 256 latent units as well as β values of 1, 10, 25, 50, and 100 was evaluated. The other hyperparameters were set to their optimized values, as applicable. We trained ten Gaussian, Bernoulli, and RBM models each in this way for 300 epochs, which allowed model weights to converge to their final values, and averages of the resultant F1 scores are displayed as heatmaps in figure 6. The figure demonstrates the performance degradation that occurs when nonoptimal hyperparameters are used. A value of the hyperparameter β of 50 or 100 is associated with moderate performance (F1 score between 0.314 and 0.536 across all three models), while a β value of 1, 10, or 25 leads to poor performance (F1 score between 0.202 and 0.328). The influence of the hyperparameter β on model performance is nonlinear, with a value of 1 or 25 resulting in better performance than a β of 10. Performance differences due to different numbers of latent units are less pronounced. All latent-space dimensions investigated in this experiment (32, 64, 128, and 256) produce good performance and correspond to the latent-space dimensions employed in all studies described in this paper (see table 1).

[10] The experiment presented in [41] was performed on an extended version of the baseline dataset of departing flights with drop-in-airspeed anomaly that comprised 20 input features, and, in contrast to the models used for the studies conducted for this paper, highly correlated features were routed through five separate encoders and decoders, and the encoder and decoder outputs were combined in, respectively, the latent space and the reconstructed input space. In the experiments reported in both papers, hyperparameters were tuned on validation sets.



(a) Gaussian (b) Bernoulli (c) RBM

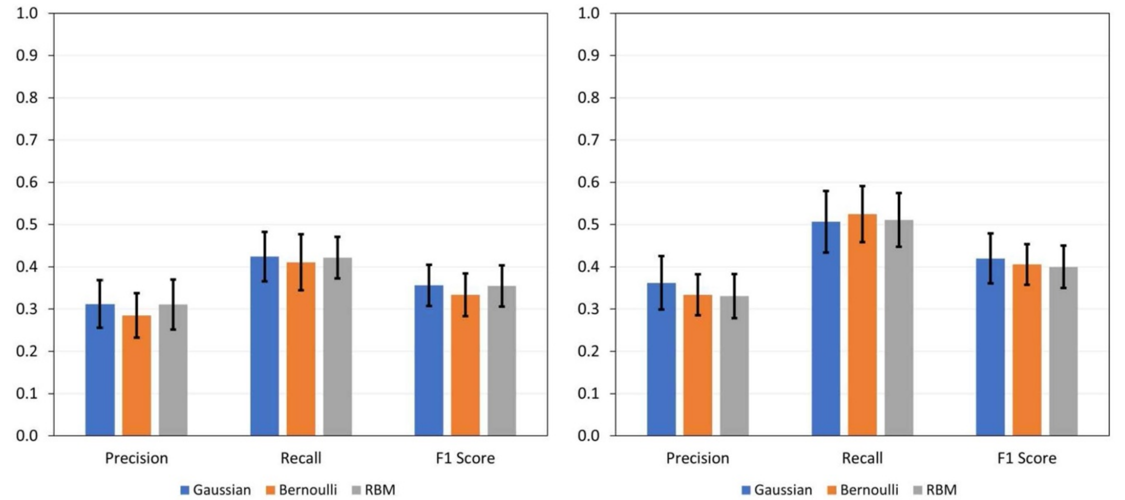
Figure 6. Performance of the Gaussian, Bernoulli, and RBM models in the baseline study after training with nonoptimal hyperparameters. The figure shows heatmaps of average F1 scores for the three models across the hyperparameters latent-space dimensionality and hyperparameter β, which are common to all three models. Across the three models, F1 scores are noticeably lower than after training with hyperparameters tuned based on validation-set performance, demonstrating the performance degradation incurred by using nonoptimal hyperparameters.

5.2. Model transferability

The DASHlink dataset[11] used in this experiment consists of time series of sensor data collected during the first 60 s of the takeoff phase of commercial flights. This is a dataset collected independently of the dataset used in the baseline study (section 5.1) but containing the same flight-parameter time series as the baseline dataset. The dataset also contains the same anomaly (drop in airspeed during takeoff by more than 20 knots) as the baseline dataset. The transferability study tests the ability of the models tuned and trained on the baseline data to transfer to another dataset containing the same input attributes and anomaly. All transferability experiments were conducted with hyperparameters determined by training on the baseline dataset's training set and assessing model performance on the baseline dataset's validation set (table 1). We examined two versions of transferability: a strong version, in which a model trained on the baseline data was directly tested on the DASHlink takeoff data, and a relaxed version, in which the best-performing model trained on the baseline data, as evaluated on the baseline data's validation set, was post-trained on the DASHlink takeoff data for 300 epochs. This method ensures that the biases and weights of the previously trained model are transferred to the target model, to be trained on the new dataset.

The DASHlink takeoff dataset of 2681 instances with 56 (2.09%) anomalies was split into training (50%) and test (50%) sets in a random but stratified way. The training set was used to determine the anomaly-score threshold [(13)] and the test set to compute average log anomaly scores [log of (10)], which were used, in conjunction with the known data labels, to determine model performance. Results are shown in figure 7. The models with Gaussian, Bernoulli, and RBM priors performed similarly; all results are within each other's error bounds (standard deviations of performance metrics among separately trained models). The average F1 scores of the Gaussian, Bernoulli, and RBM models are 0.356, 0.334, and 0.355, respectively, in the transferability experiment without post-training (figure 7(a)). The values of this metric (in the same order) for the experiment with post-training of the best-performing baseline model are 0.420, 0.406, and 0.400 (figure 7(b)). Comparing the baseline experiments and the transferability experiments with post-training, the model transfer reduced anomaly-detection performance (as measured by F1 score averaged across models) by 20.0 percentage points. Post-training on the target dataset slightly improves the overall F1-score average, by 6.03 percentage points. The relatively small extent of the performance improvement highlights the importance of hyperparameters for model performance. The structure of the results is similar to that in the baseline study, with recall higher than precision. The Bernoulli model does not lag behind the other models, as in the baseline study. Overall, the model transfer markedly reduced model performance while still producing usable results. The transferability results are more difficult to interpret with regard to performance differences between the three models than the baseline results because the experiments from which they were obtained were more complex and involved additional factors (compared with the baseline experiments) since they involved two datasets. We report additional transferability experiments in supplementary section S2.

[11] extracted from the data posted at https://c3.ndc.nasa.gov/dashlink/projects/85/ [77].



(a) Transferability experiment with DASHlink threshold and without post-training

(b) Transferability experiment with DASHlink threshold and with post-training

Figure 7. Performance of VAE models with Gaussian, Bernoulli, and RBM priors in the study examining the ability of the models tuned and trained on the baseline data to transfer to the DASHlink takeoff data. Error bars indicate standard deviations of performance metrics among 16 independently trained models. Performance was assessed based on a threshold derived from the DASHlink training set, (a) without or (b) with post-training on the new dataset for 300 epochs with model weights initialized by training to convergence on the baseline dataset. The figure depicts similar performance for all models, trained without supervision, markedly worse than in the baseline experiment, but still in the useful range.

5.3. Robustness of RBM model: delay in flap deployment during approach to landing

To deepen our insight into the behavior and performance of the quantum-compatible DVAE model with RBM prior, we conducted a further anomaly-detection study with this model, on another dataset, including *de novo* determination of hyperparameters and training. The experimental approach for this study is similar to the one adopted for the baseline study (section 5.1). The anomaly in the dataset used for this study is a delay in the deployment of flaps, as judged by a subject matter expert, during the final approach to landing, lasting approximately 160 s[12]. As in the previous experiments, this labeled anomaly is not necessarily the only anomaly in the dataset, and the label is not among the model inputs during the unsupervised training. The dataset contains ten time series of aeronautical sensor outputs, relating to position, orientation, and speed of the aircraft, as well as to the positions of the control surfaces.

The dataset was randomly divided into training (60%), validation (20%), and test (20%) sets. Since the data were normalized to lie in the range of zero to one using (11), we used the BCE [(12)] to estimate the reconstruction error [negative of left term in (6)]. Training proceeded for 400 epochs and weights converged to their final values after about 100 epochs. We again independently trained 16 models and report the mean +/− standard deviation of their performance. We did not observe any overfitting over the epochs of training, as visualized by the progression of the losses of the validation set. The hyperparameters tried out that produced the best overall performance in precision, recall, and F1 score, evaluated on the validation set, are given in table 1 and were used for this study. To test model performance, the training and validation sets were combined for the purpose of model training, and then the performance of the models trained in this way was evaluated on the test set. It should be noted that in this study anomaly scores were not log-transformed because the original scores were already approximately normally distributed. Log transformation neither increased the scores' normality nor improved anomaly-detection performance. Additional information is provided in supplementary section S3.

Table 3 shows the performance of the RBM model in this case study. The model achieved a mean precision of 0.591, a mean recall of 0.647, and a mean F1 score of 0.618. The mean F1 score is similar to (slightly lower than) the F1 value of 0.666 achieved by the RBM model in the baseline study, which was

[12] The dataset contains 21 302 instances, 954 (4.48%) of which exhibit the delay-in-flap-deployment anomaly. A larger curated unnormalized dataset containing these data is publicly available at https://c3.ndc.nasa.gov/dashlink/resources/1018/ [78]. The dataset used for the study can be reproduced from the posted dataset, up to stochastic differences due to the random selection of data points. To replicate the study's data, one (randomly) chooses a subset of the nominal data, chooses only the anomalous data with the pertinent anomaly, and scales the data according to (11), with ξ referring to the training data. Only the following attributes are then included: Corrected AOA, Barometric Altitude, Computed Airspeed, TE Flap Position, Glideslope Deviation, Core Speed AVG, Pitch Angle, Roll Angle, True Heading, and Wind Speed.

Table 3. Performance of the VAE model with RBM prior in the experiment with late-deployment-of-flaps anomaly during final approach to landing.

Performance metric	Average	Standard deviation
Precision	0.591	0.0543
Recall	0.647	0.0564
F1 Score	0.618	0.0554

performed on a dataset with drop-in-airspeed anomaly during takeoff. Consequently, the late-deployment-of-flaps study corroborates the excellent anomaly-detection performance of the unsupervised DVAE model with RBM prior observed in the baseline study and illustrates the model's robustness to change of anomaly type and phase of flight.

5.4. Model design

We developed two DVAE models. The Bernoulli model employs a factorized Bernoulli distribution as prior, in analogy to the predominant continuous VAE with Gaussian prior [(1)]. The RBM model is an attempt to make the DVAE more flexible and expressive. It places an RBM in the VAE's latent space, so that the VAE's latent units are the RBM's inputs, and uses the energy-based restricted Boltzmann distribution given in (2) as prior. The RBM model parameters **a**, **b**, and **W** are tuned during training by the interplay between positive- and negative-phase energies, and the 'visible' and 'hidden' units of the negative phase (fantasy states $\tilde{\mathbf{z}}_v^d$ and $\tilde{\mathbf{z}}_h^d$, respectively) are updated via MCMC Gibbs sampling from (3).

DVAE models are less common than VAEs with continuous latent variables because an ELBO objective [(4)] containing discrete variables cannot be differentiated, thus precluding the computation of ELBO gradients, a necessary operation during the backward pass of training. In order to make DVAEs differentiable, the reparameterization trick [10], which moves the variational parameters $\boldsymbol{\phi}$ from the distribution q_ϕ to a more easily differentiable deterministic function $g_{\boldsymbol{\phi}}$, is extended by a smoothing function [26] or a continuous relaxation [24, 60]. Our Bernoulli and RBM models employ the Gumbel-softmax trick [24, 60] to relax the discrete posterior latents $\mathbf{z}^d \sim q_\phi(\mathbf{z}^d|\mathbf{x})$.

Several authors [26, 29–31] break up the VAE's latent space into two equally sized partitions and implement the positive phase of the RBM between the two halves of latent units **z** sampled from the approximate posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ and, similarly, the negative phase between the latent units $\tilde{\mathbf{z}}^d$ sampled from the RBM prior $p_{\boldsymbol{\theta}}(\tilde{\mathbf{z}}^d)$. In this paper, we have termed an RBM model with such characteristics a 'bipartite latent-space RBM.' In the experiments performed for this paper, we have adopted a somewhat different approach and introduced a hidden layer in the RBM's positive phase, which is obtained in one Gibbs-sampling step from the positive phase's visible layer, comprising the VAE's latent units. The evolution of the negative phase's fantasy states is accomplished by k-step Gibbs sampling ($k \geqslant 1$) in both RBM versions (see figure 2).

In addition to adding a truly hidden layer to the RBM's positive phase, we also introduced the hyperparameter β into the ELBO objective function [see (7)]. While Higgins *et al* [21] devised the β-ELBO, in which β regulates the trade-off between the autoencoding and KL-divergence terms, as a way to promote disentanglement between latent dimensions by enforcing a minimum similarity between variational posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ and prior $p_{\boldsymbol{\theta}}(\mathbf{z})$, we tune β to optimize anomaly-detection performance, as measured by performance metrics [(14)].

5.5. Hyperparameters and normalizing transformations

Latent-space dimensionality and the hyperparameter β are hyperparameters that have a strong effect on model performance (in all three models) and require optimization for each application. Models with 32–256 latent units generally exhibited good performance, whereas model performance was very sensitive to the specific choice of β; hence, the choice of the value of the hyperparameter β is application-specific. The temperature λ of the concrete distribution [24, 60], which determines differentiability with respect to the variational parameters and the bias in gradient estimation introduced by the continuous Gumbel-softmax relaxation, is also important for the discrete models (Bernoulli and RBM) and requires tuning. We used a λ of 0.1 throughout our studies because it balances the trade-off between estimation bias and differentiability and had proved the optimal value among the ones tested. We did not use an annealing schedule to gradually reduce λ over the epochs of training, but we used discrete ('hard') Bernoulli variables for validation and testing. While the length of the persistent chains (fantasy particles) had a mild influence on the performance of the RBM model, the number of fantasy particles was quite unimportant, due to the averaging of the particles' (negative-phase) energies at the end of each series of Gibbs updates (per minibatch); we used 500 throughout. When optimal, or even adequate, hyperparameters are unknown and labeled data not available, which is often the case in practical applications, the performance of unsupervised anomaly-detection models can frequently be improved by making the effort to prepare a small labeled validation set that includes relevant anomalies and using it for hyperparameter tuning [68, 69].

We were able to boost anomaly-detection performance by applying a normalizing transformation to skewed anomaly scores (MSEs) with a conspicuous right tail. We always used a log transformation because, on average, it had performed best on validation data, but less skewed scores might benefit more from a square-root transformation and for more strongly skewed scores an inverse transformation might be optimal.

6. Conclusions

The accurate and timely discovery of flight-operation anomalies is important because they can be precursors of potentially serious aviation incidents or accidents. To detect operationally significant anomalies and preempt future accidents, airlines and transportation agencies will have to increasingly rely on advanced machine-learning techniques applied to historical data or online data streams. Multifactorial and nonlinear anomalies defy traditional anomaly-detection methods, such as exceedance detection [70, 71], and the volume and proportion of anomalies characterized by heterogeneous and high-order feature interactions are only expected to grow with increasing airspace complexity, characterized by increasing passenger volume, the integration of unmanned aircraft systems, and urban air mobility. Since labeled data (i.e. data classified as either nominal or anomalous) are costly to obtain and frequently not available and flight anomalies heterogeneous in nature, unsupervised learning approaches, such as the one portrayed in this paper, are often the preferred or only feasible option.

We developed unsupervised convolutional VAE models with a latent space based on Gaussian, Bernoulli, and RBM priors. The discrete Bernoulli and RBM versions are an attempt to design models whose latent space captures the discrete nature of objects processed by machine-learning models, such as semantic classes, causal factors, digital samples, and other discrete entities. The RBM model allows a straightforward integration with quantum computing because the fantasy states of the RBM's negative phase can be sampled by measuring a parameterizable density operator $\rho_{\boldsymbol{\theta}}$ or wavefunction $\psi_{\boldsymbol{\theta}}$ in the computational basis. Such a quantum generative process can be produced by a quantum Boltzmann machine (QBM) implemented on an annealer [61] or by a quantum-circuit Born machine (QCBM) [72].

While the developed models exhibit good performance overall, the discrete Bernoulli model performs more poorly than the other two. On the other hand, the more expressive RBM model, which, during training, employs unnormalized energies rather than probabilities and MCMC to sample from the prior distribution, is a discrete-variable model whose performance matches that of the VAE model with continuous Gaussian prior, the customarily employed prior type. Transferring a model without re-tuning of hyperparameters or re-training to a new dataset with the same anomaly results in an anomaly-detection performance that is noticeably impaired, but still respectable, considering that the datasets involved consist of multiple complex time series and the training was executed without supervision. Moreover, the model with RBM prior proved robust to changes in the type of anomaly and phase of flight. We would like to note that our VAE models with continuous and discrete priors are universal generative models and not limited in application to aeronautics data and, with slight modification, can also be applied to other time-series or image data.

In future studies, we will use more advanced algorithms, such as parallel tempering and adaptive-step PCD [50, 54, 73–76], to conduct negative-phase sampling, to see if such schemes improve model performance. We will also devise an estimator of the log partition function, to compute log-likelihood estimates for future generation studies. We also intend to use conditional VAEs or other deep generative models to generate artificial anomalies, to enhance anomaly-detection datasets and, ultimately, the performance of anomaly-detection algorithms. We plan to integrate quantum capabilities by developing a QBM, implemented by quantum annealing, or a QCBM. In addition, we intend to further increase the expressiveness of the VAE prior by replacing the relatively simple RBM network in the latent space with a more sophisticated energy-based feedforward network, and we will attempt to integrate such an EBM with quantum computing.

Data availability statement

The data that support the findings of this study are openly available at the following URLs/DOIs: https://c3.ndc.nasa.gov/dashlink/projects/85/ [77] and https://c3.ndc.nasa.gov/dashlink/resources/1018/ [78]. The proprietary data used for the baseline study (section 5.1) cannot be made available for legal reasons. The datasets from which the data used for the studies on model transferability (section 5.2) and the robustness of the RBM model (section 5.3) were extracted are available at, respectively, https://c3.ndc.nasa.gov/dashlink/projects/85/ [77] and https://c3.ndc.nasa.gov/dashlink/resources/1018/ [78] (see footnote 12 for details). The software source code implementing our deep-learning models and containing the scripts to train them and evaluate their performance on datasets is under official review for software-release authorization at the time of publication and will be made available at NASA GitHub (https://github.com/nasa/) once approval of release has been obtained.

Acknowledgments

We would like to thank Bryan Matthews for his help with preparing the datasets used in the studies reported in this paper and Arash Vahdat for a helpful discussion on energy-based modeling while preparing this work. This work was supported by NASA Ames Center Innovation Funding. A A A, M M, and P A L acknowledge funding support from NASA Academic Mission Services Contract NNA16BD14C (System-Wide Safety Project). A A acknowledges funding support from NASA Academic Mission Services Contract NNA16BD14C and from NSF Award CCF-1918549 through the Feynman Quantum Academy Internship Program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon.

Appendix A. Generative model (decoder) and recognition model (encoder) of directed generative model with latent variables

The generative model (decoder) reconstructs the input data from the latent variables [5, 55]. The statistics of the generative model's output are given by the marginal distribution:

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int_{\mathbf{z}} p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})\mathrm{d}\mathbf{z} = \int_{\mathbf{z}} p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{z})\mathrm{d}\mathbf{z}. \tag{A1}$$

Here, the joint distribution of the input variables $\mathbf{x}$ and the latent variables $\mathbf{z}$, $p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})$, defines the model distribution. In a directed generative model, the model distribution is explicitly factored into the generative distribution $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$, the distribution of the generative model's output given a latent-space realization, and the model's prior distribution $p_{\boldsymbol{\theta}}(\mathbf{z})$. The marginal distribution is then obtained by integrating ('marginalizing') over the latent variables (see figure 1). If the latent variables are discrete, the integration in (A1) is replaced by summation.

In representational learning, input-dependent latent variables (the 'code') are identified by a second model, the recognition model (encoder). The lower the number of latent variables, the more compressed is the model's latent space, with the degree of compression given by the ratio of input to latent variables. The statistical distribution of the recognition model's output is called the recognition or posterior distribution, $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$, the probability of the set of latent variables conditioned on the input data (see figure 1). The posterior distribution is given by Bayes' rule as [55]:

$$p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x})} = \frac{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x})}. \tag{A2}$$

Appendix B. Variational inference

Models that allow the tractable computation of the posterior distribution are called invertible and those that do not noninvertible [5]. Noninvertible models do not allow gradient computations and concomitant optimization. Deep latent-variable models are noninvertible because no analytic solution or efficient estimation procedure exists for the marginal probability given in (A1). Since the marginal is intractable, the posterior given in (A2) is as well as it requires the marginal in its denominator [55]. Approximate-inference techniques approximate the true posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ and marginal $p_{\boldsymbol{\theta}}(\mathbf{x})$ [5, 55]. The approximate posterior is written as $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ (see figure 1). Variational inference is such an approximation technique. In variational inference, the marginal log likelihood is decomposed as follows [10, 55]:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})) + \mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}). \tag{B1}$$

The expression $D_{\mathrm{KL}}(p||q) \equiv \mathbb{E}_p \log[p/q]$ stands for the Kullback-Leibler (KL) divergence, an asymmetric 'distance' between two probability distributions. The term $\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x})$ denotes the ELBO—a variational approximation, from below, to the true marginal log likelihood. As the KL divergence is non-negative, the

ELBO is the greater and closer to the log likelihood, i.e. the bound tighter, the closer the approximating posterior to the true posterior. Hence, maximizing the ELBO simultaneously increases the (log) likelihood and decreases the distance between the variational and true posteriors.

Appendix C. Unbiased gradients with respect to generative and variational parameters of VAE with factorized Gaussian prior

Unbiased gradients with respect to the generative parameters $\boldsymbol{\theta}$ are straightforward to obtain, and we can write:

$$\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[f(\mathbf{z})]=\mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\nabla_{\boldsymbol{\theta}}f(\mathbf{z})]\simeq\frac{1}{S}\sum_{s=1}^{S}\nabla_{\boldsymbol{\theta}}f(\mathbf{z}^{(s)}). \tag{C1}$$

The mean on the right of (C1) is a Monte Carlo estimate of the gradient with respect to $\boldsymbol{\theta}$, in which $\mathbf{z}^{(s)}$ is an i.i.d. sample from $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ and S indicates the size of the minibatch.

Unbiased gradients with respect to the variational parameters $\boldsymbol{\phi}$ are more difficult to obtain because the expectations in (4) are estimated by sampling from a probability distribution that depends on $\boldsymbol{\phi}$ [29]. Based on the gradient of the logarithm of $q_{\boldsymbol{\phi}}$, $\nabla_{\boldsymbol{\phi}}\log q_{\boldsymbol{\phi}}$, the score-function estimator [79–81] calculates gradients with respect to $\boldsymbol{\phi}$ as:

$$\begin{aligned}\nabla_{\boldsymbol{\phi}}\mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[f(\mathbf{z})]&=\mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[f(\mathbf{z})\nabla_{\boldsymbol{\phi}}\log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})]\\&\simeq\frac{1}{S}\sum_{s=1}^{S}f(\mathbf{z}^{(s)})\nabla_{\boldsymbol{\phi}}\log q_{\boldsymbol{\phi}}(\mathbf{z}^{(s)}|\mathbf{x})).\end{aligned} \tag{C2}$$

Here, we assumed for simplicity that f does not depend on $\boldsymbol{\phi}$. Unfortunately, gradients based on the score-function estimator are characterized by high variance and require the use of intricate variance-reduction techniques (such as control variates) in practical applications [56, 82].

The reparameterization trick [10] is used in VAEs as a low-variance alternative to the score-function estimator. Here, the random variable $\mathbf{z}\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ is re-expressed by means of an auxiliary random variable $\boldsymbol{\epsilon}\sim p(\boldsymbol{\epsilon})$ independent of $\boldsymbol{\phi}$ and a deterministic function $g_{\boldsymbol{\phi}}(\cdot)$ as $\mathbf{z}=g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon},\mathbf{x})$. We then can write $\mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[f(\mathbf{z})]=\mathbb{E}_{\boldsymbol{\epsilon}\sim p(\boldsymbol{\epsilon})}[f(g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon},\mathbf{x}))]$ and obtain unbiased Monte Carlo estimates of the gradient with respect to $\boldsymbol{\phi}$ by moving the gradient operator into the expectation:

$$\begin{aligned}\nabla_{\boldsymbol{\phi}}\mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[f(\mathbf{z})]&=\mathbb{E}_{\boldsymbol{\epsilon}\sim p(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\phi}}f(g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon},\mathbf{x}))]\\&\simeq\frac{1}{S}\sum_{s=1}^{S}\nabla_{\boldsymbol{\phi}}f(g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(s)},\mathbf{x})),\end{aligned} \tag{C3}$$

where $\boldsymbol{\epsilon}^{(s)}\sim p(\boldsymbol{\epsilon})$. The reparameterization trick transfers the dependence on $\boldsymbol{\phi}$ from $q_{\boldsymbol{\phi}}$ into f, substituting the problem of estimating the gradient with respect to the variational parameters of a distribution with the simpler problem of estimating the gradient with respect to the variational parameters of a deterministic function [60].

In a VAE with a factorized Gaussian prior, the latent variables produced by the encoder $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})=\mathcal{N}(\mathbf{z};\boldsymbol{\mu},\mathrm{diag}(\boldsymbol{\sigma}^2))\equiv\prod_l\mathcal{N}(z_l;\mu_l,\sigma_l^2))$ can be reparameterized as $\mathbf{z}=\boldsymbol{\mu}+\boldsymbol{\sigma}\odot\boldsymbol{\epsilon}$, where $\odot$ represents the elementwise product and $\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})$ [55]. In other words, the components of the reparameterized latent vector $\mathbf{z}$ are reparameterized univariate Gaussians $z_l=\mu_l+\sigma_l\epsilon_l$, $\epsilon_l\sim\mathcal{N}(0,1)$. Unbiased gradients with respect to the variational parameters $\boldsymbol{\phi}$ are then obtained as:

$$\begin{aligned}\nabla_{\boldsymbol{\phi}}\mathbb{E}_{\mathbf{z}\sim\mathcal{N}(\boldsymbol{\mu},\mathrm{diag}(\boldsymbol{\sigma}^2))}[f(\mathbf{z})]&=\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[\nabla_{\boldsymbol{\phi}}f(\boldsymbol{\mu}+\boldsymbol{\sigma}\odot\boldsymbol{\epsilon})]\\&\simeq\frac{1}{S}\sum_{s=1}^{S}\nabla_{\boldsymbol{\phi}}f(\boldsymbol{\mu}+\boldsymbol{\sigma}\odot\boldsymbol{\epsilon}^{(s)}).\end{aligned} \tag{C4}$$

Appendix D. Derivation of β-ELBO

The constrained optimization problem for a VAE is specified in (D1), where δ controls the strength of the applied constraint.

$$\begin{aligned} &\max_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] \\ &\text{s.t.} \quad D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z})) < \delta \end{aligned}. \tag{D1}$$

Using KKT conditions [83, 84] to re-write (D1) as a Lagrangian yields:

$$\mathcal{F}(\boldsymbol{\theta},\boldsymbol{\phi},\beta;\mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] - \beta(D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z})) - \delta), \tag{D2}$$

where the KKT multiplier β is a regularization coefficient that constrains latent-space capacity and exerts implicit pressure on the latent-space variables $\mathbf{z}$, which represent the input data $\mathbf{x}$, to become less correlated by drawing each component variable z_l in the direction of the corresponding variable sampled from the prior. Higgins *et al* [21] demonstrate that a higher β leads to less entangled latent variables, but it also decreases reconstruction quality. Disentanglement is easy to visualize in images, by observing the continuous change of a factor when a latent dimension is varied while the others are held constant. By eliminating the δ term, (D2) can be written as a β-ELBO:

$$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x},\beta) = \mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] - \beta D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z})). \tag{D3}$$

Since both β and δ are non-negative constants in (D2), $\mathcal{L}$ bounds $\mathcal{F}$ from below: $\mathcal{F}(\boldsymbol{\theta},\boldsymbol{\phi},\beta;\mathbf{x}) \geqslant \mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x},\beta)$.

Appendix E. Reparameterization trick and unbiased gradients with respect to variational parameters of VAE with discrete latent space

If each individual latent variable is sampled as $z^d \sim \mathcal{B}(p)$, where $\mathcal{B}$ denotes the Bernoulli distribution and p its parameter, the concrete relaxation can be expressed as [60]:

$$\mathbf{z} = \sigma((\log\boldsymbol{\alpha} + \log\boldsymbol{\rho} - \log(1-\boldsymbol{\rho}))/\lambda), \tag{E1}$$

where σ denotes the logistic function, $\sigma(x) = 1/(1+e^{-x})$, α the odds of the Bernoulli probability, $\alpha = p/(1-p)$, ρ a continuous uniform random variable on the interval [0, 1], and λ the temperature of the concrete distribution. The temperature parameter λ controls the degree of continuous relaxation: the greater λ, the greater is the relaxation and departure from a Bernoulli random variable, whereas small λ values produce continuous variates close to 0 and 1. In our studies, we consistently use a λ of 0.1, which we determined as a hyperparameter with good performance and which introduces only a small bias in gradient estimation. The concrete or Gumbel-softmax relaxation was only applied during training and not during evaluation (validation or testing) because differentiability of the objective function is only required during the training phase and we sought to retain, where possible, the discrete character of the model and avoid the (slight) estimation bias introduced by the continuous relaxation.

Unbiased gradients of the posterior latents with respect to the variational parameters $\boldsymbol{\phi}$ of the encoder are then obtained as:

$$\begin{aligned} \nabla_{\boldsymbol{\phi}}\mathbb{E}_{\mathbf{z}\sim\mathcal{B}(\mathbf{q})}[f(\mathbf{z})] &= \mathbb{E}_{\boldsymbol{\rho}\sim\mathcal{U}(\mathbf{0},\mathbf{1})}[\nabla_{\boldsymbol{\phi}} f(\sigma((\log\boldsymbol{\alpha}^{\mathbf{q}} + \log\boldsymbol{\rho} - \log(1-\boldsymbol{\rho}))/\lambda))] \\ &\simeq \frac{1}{S}\sum_{s=1}^{S}\nabla_{\boldsymbol{\phi}} f(\sigma((\log\boldsymbol{\alpha}^{\mathbf{q}} + \log\boldsymbol{\rho}^{(s)} - \log(1-\boldsymbol{\rho}^{(s)}))/\lambda)), \end{aligned} \tag{E2}$$

where $\mathcal{B}(\mathbf{q})$ denotes the *relaxed* Bernoulli distribution with parameter $\mathbf{q}$, $\log\boldsymbol{\alpha}^{\mathbf{q}}$ the log-odds parameter vector of the relaxed Bernoulli approximate posterior probabilities, and $\boldsymbol{\rho}\sim\mathcal{U}(\mathbf{0},\mathbf{1})$.

Appendix F. KL-divergence terms in ELBO objective of DVAE models

F.1. Bernoulli prior

F.1.1. Stochastic expression

Based on the probability mass function of a Bernoulli random variable, $\mathcal{B}(k;p) = p^k\,(1-p)^{1-k}$, the KL term of the ELBO of a VAE with a Bernoulli prior can be expressed as:

$$\begin{aligned}
D_{\mathrm{KL}}(q_\phi(\mathbf{z}^d|\mathbf{x})||p_\theta(\mathbf{z}^d)) &= \mathbb{E}_{\mathbf{z}^d\sim q_\phi(\mathbf{z}^d|\mathbf{x})}\left[\log\frac{q_\phi(\mathbf{z}^d|\mathbf{x})}{p_\theta(\mathbf{z}^d)}\right] \\
&= \mathbb{E}_{\mathbf{z}^d\sim\mathcal{B}(\mathbf{q})}\left[\sum_{l=1}^{L}\log\frac{q_l^{z_l^d}\,(1-q_l)^{1-z_l^d}}{0.5^{z_l^d}\,(1-0.5)^{1-z_l^d}}\right] \\
&= \mathbb{E}_{\mathbf{z}^d\sim\mathcal{B}(\mathbf{q})}\left[\sum_{l=1}^{L}(z_l^d\log q_l + (1-z_l^d)\log(1-q_l) - z_l^d\log 0.5 - (1-z_l^d)\log(1-0.5))\right] \\
&= \mathbb{E}_{\mathbf{z}^d\sim\mathcal{B}(\mathbf{q})}\left[\sum_{l=1}^{L}\left(-z_l^d\log\frac{0.5}{q_l} - (1-z_l^d)\log\frac{1-0.5}{1-q_l}\right)\right],
\end{aligned} \tag{F1}$$

where q_l stands for the parameter of the latent Bernoulli variable $z_l^d \sim \mathcal{B}(q_l)$, considered to be *discrete*, and 0.5 is the parameter of the Bernoulli prior distribution, $\mathcal{B}(p_l = 0.5)$. The KL term can be implemented with a binary-cross-entropy loss with logits, frequently available as a class or function in deep-learning libraries, with the log-odds parameter vector of the respective *relaxed* Bernoulli distribution [$\log\boldsymbol{\alpha}^{\mathbf{q}}$ (approximate posterior) or $\log\boldsymbol{\alpha}^{\mathbf{p}} = \mathbf{0}$ (prior)] and the *relaxed* latent vector $\mathbf{z}$ as input arguments. We used this (stochastic) method [Monte Carlo estimator of (F1)] in all experiments in which the VAE had a Bernoulli prior.

F.1.2. Analytic expression

For a Bernoulli prior, the KL term can also be written as:

$$D_{\mathrm{KL}}(\mathcal{B}(\mathbf{z}^d;\mathbf{q})||\mathcal{B}(\mathbf{z}^d;\mathbf{0.5})) = \sum_{l=1}^{L}\sum_{k=0}^{1} q_l^k\,(1-q_l)^{1-k}\log\frac{q_l^k\,(1-q_l)^{1-k}}{0.5^k\,(1-0.5)^{1-k}}, \tag{F2}$$

where the possible outcomes k in the support of the posterior distribution $q_\phi(\mathbf{z}^d|\mathbf{x})$ are considered to be *discrete* and l indexes a latent variable. Algebraic manipulation of (F2) then leads to an alternative analytic expression for the KL term of a VAE with Bernoulli prior:

$$D_{\mathrm{KL}}(\mathcal{B}(\mathbf{z}^d;\boldsymbol{\alpha}^{\mathbf{q}})||\mathcal{B}(\mathbf{z}^d;\mathbf{1})) = \sum_{l=1}^{L}\frac{\alpha_l^q\log(\frac{\alpha_l^q}{\alpha_l^q+1}) + \log(\frac{1}{\alpha_l^q+1}) + (\alpha_l^q+1)\log 2}{\alpha_l^q+1}, \tag{F3}$$

where $\boldsymbol{\alpha}^{\mathbf{q}}$ stands for the vector of the odds of the parameters of the Bernoulli posterior and reparameterizes it. We tested this analytic method using the odds of the *relaxed* Bernoulli posterior to compute the KL-divergence term in the minibatch-based β-ELBO loss [negative of (7) over minibatches] during training, and it basically produces the same results as the stochastic approximation described in the previous section (same results within the bounds of stochastic variability).

F.2. RBM prior (stochastic)

Since the applied concrete (Gumbel-softmax) relaxation replaces discrete latent variables $\mathbf{z}^d$ with continuous variables $\mathbf{z}$, the KL term of the ELBO of a VAE with a relaxed Bernoulli approximate posterior and an RBM prior can be expressed as:

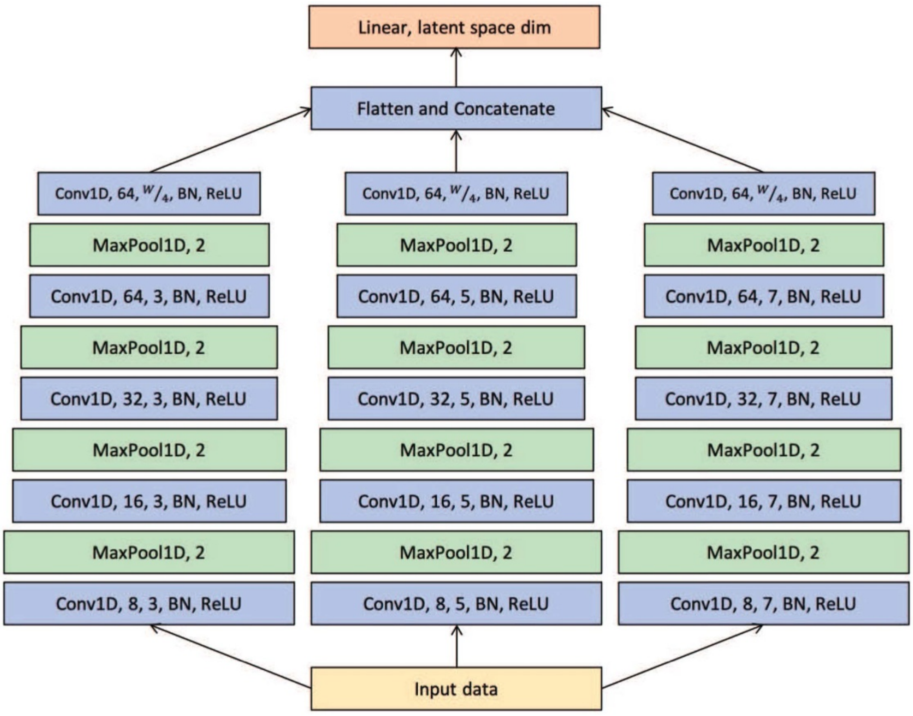


Figure G1. Architecture of the encoder of the VAE models.

$$\begin{aligned}
&D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}))\\
&\quad=\mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z})}\right]\\
&\quad=\mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})]-\mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{z})]\\
&\quad=\mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})]-\mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\frac{\mathrm{e}^{-E_{\boldsymbol{\theta}}(\mathbf{z})}}{Z_{\boldsymbol{\theta}}}\right]\\
&\quad=\mathbb{E}_{\boldsymbol{\rho}\sim\mathcal{U}(\mathbf{0},\mathbf{1})}[\log q_{\boldsymbol{\phi}}(\mathbf{z}(\boldsymbol{\phi},\boldsymbol{\rho})|\mathbf{x})]-\mathbb{E}_{\boldsymbol{\rho}\sim\mathcal{U}(\mathbf{0},\mathbf{1})}\left[\log\frac{\mathrm{e}^{-E_{\boldsymbol{\theta}}(\mathbf{z}(\boldsymbol{\phi},\boldsymbol{\rho}))}}{Z_{\boldsymbol{\theta}}}\right]\\
&\quad=\mathbb{E}_{\boldsymbol{\rho}\sim\mathcal{U}(\mathbf{0},\mathbf{1})}[\log q_{\boldsymbol{\phi}}(\mathbf{z}(\boldsymbol{\phi},\boldsymbol{\rho})|\mathbf{x})]-\mathbb{E}_{\boldsymbol{\rho}\sim\mathcal{U}(\mathbf{0},\mathbf{1})}[\log\mathrm{e}^{-E_{\boldsymbol{\theta}}(\mathbf{z}(\boldsymbol{\phi},\boldsymbol{\rho}))}]+\log Z_{\boldsymbol{\theta}}\\
&\quad=\mathbb{E}_{\boldsymbol{\rho}\sim\mathcal{U}(\mathbf{0},\mathbf{1})}[\log q_{\boldsymbol{\phi}}(\mathbf{z}(\boldsymbol{\phi},\boldsymbol{\rho})|\mathbf{x})]+\mathbb{E}_{\boldsymbol{\rho}\sim\mathcal{U}(\mathbf{0},\mathbf{1})}[E_{\boldsymbol{\theta}}(\mathbf{z}(\boldsymbol{\phi},\boldsymbol{\rho}))]+\log Z_{\boldsymbol{\theta}}.
\end{aligned}\tag{F4}$$

Since $\nabla_{\boldsymbol{\theta}}\log Z_{\boldsymbol{\theta}}=-\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\tilde{\mathbf{z}}^d\sim p_{\boldsymbol{\theta}}(\tilde{\mathbf{z}}^d)}[E_{\boldsymbol{\theta}}(\tilde{\mathbf{z}}^d)]$ [30, 85], unbiased gradients of the KL term with respect to the generative and variational parameters can be obtained as:

$$\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}}\{\mathbb{E}_{\boldsymbol{\rho}\sim\mathcal{U}(\mathbf{0},\mathbf{1})}[\log q_{\boldsymbol{\phi}}(\mathbf{z}(\boldsymbol{\phi},\boldsymbol{\rho})|\mathbf{x})]+\mathbb{E}_{\boldsymbol{\rho}\sim\mathcal{U}(\mathbf{0},\mathbf{1})}[E_{\boldsymbol{\theta}}(\mathbf{z}(\boldsymbol{\phi},\boldsymbol{\rho}))]-\mathbb{E}_{\tilde{\mathbf{z}}^d\sim p_{\boldsymbol{\theta}}(\tilde{\mathbf{z}}^d)}[E_{\boldsymbol{\theta}}(\tilde{\mathbf{z}}^d)]\}.\tag{F5}$$

The symbolic expression $\tilde{\mathbf{z}}^d$ denotes 'fantasy states,' i.e. values of the latent variables produced by the RBM model (prior) distribution, which remain discrete and are not relaxed during training [30, 31].

Appendix G. Architecture of β-CVAE models

Figure G1 shows the architecture of the encoder of the VAE models. The input data traverse three parallel branches of 1D convolution operations with different filter sizes (first numeric) and kernel sizes (second numeric), followed by batch normalization and ReLU activation, and finally max pooling of size 2. The decoder is identical to an inverted encoder in which 1D convolutions have been replaced with 1D transpose convolutions and max pooling with upsampling (with bilinear interpolation).

ORCID iD

Thomas Templin https://orcid.org/0000-0001-5936-0009

References

[1] Hinton G E, Osindero S and Teh Y W 2006 *Neural Comput.* **18** 1527–54

[2] Bengio Y, Lamblin P, Popovici D and Larochelle H 2006 *Advances in Neural Information Processing Systems* vol 19

[3] LeCun Y, Bengio Y and Hinton G 2015 *Nature* **521** 436–44

[4] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (MIT Press)

[5] Dayan P and Abbott L F 2005 *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems* (MIT Press)

[6] Witten I H, Frank E, Hall M A and Pal C J 2017 *Data Mining: Practical Machine Learning Tools and Techniques* 4th edn (Elsevier Inc.)

[7] Hinton G E, Dayan P, Frey B J and Neal R M 1995 *Science* **268** 1158–61

[8] Vincent P, Larochelle H, Bengio Y and Manzagol P A 2008 Extracting and composing robust features with denoising autoencoders *Proc. 25th Int. Conf. on Machine Learning* pp 1096–103

[9] Goodfellow I J, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2014 Generative adversarial networks (arXiv:1406.2661)

[10] Kingma D P and Welling M 2013 Auto-encoding variational Bayes (arXiv:1312.6114)

[11] Hinton G E and Salakhutdinov R R 2006 *Science* **313** 504–7

[12] Goodfellow I 2016 NIPS 2016 tutorial: generative adversarial networks (arXiv:1701.00160)

[13] Dumoulin V, Belghazi I, Poole B, Mastropietro O, Lamb A, Arjovsky M and Courville A 2016 Adversarially learned inference (arXiv:1606.00704)

[14] Donahue J, Krähenbühl P and Darrell T 2016 Adversarial feature learning (arXiv:1605.09782)

[15] Zenati H, Romain M, Foo C S, Lecouat B and Chandrasekhar V 2018 Adversarially learned anomaly detection *2018 IEEE Int. Conf. on Data Mining* (*ICDM*) (IEEE) pp 727–36

[16] Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A and Chen X 2016 *Advances in Neural Information Processing Systems* vol 29

[17] Arjovsky M and Bottou L 2017 Towards principled methods for training generative adversarial networks (arXiv:1701.04862)

[18] Arjovsky M, Chintala S and Bottou L 2017 Wasserstein generative adversarial networks *Int. Conf. on Machine Learning* (PMLR) pp 214–23

[19] Wiatrak M, Albrecht S V and Nystrom A 2019 Stabilizing generative adversarial networks: a survey (arXiv:1910.00927)

[20] Burda Y, Grosse R and Salakhutdinov R 2015 Importance weighted autoencoders (arXiv:1509.00519)

[21] Higgins I, Matthey L, Pal A, Burgess C, Glorot X, Botvinick M Mohamed S and Lerchner A 2017 Beta-VAE: learning basic visual concepts with a constrained variational framework *Proc. Int. Conf. on Learning Representations* (*ICLR*) pp 24–26

[22] Burgess C P, Higgins I, Pal A, Matthey L, Watters N, Desjardins G and Lerchner A 2018 Understanding disentangling in β-VAE (arXiv:1804.03599)

[23] Kingma D P, Mohamed S, Jimenez Rezende D and Welling M 2014 *Advances in Neural Information Processing Systems* vol 27

[24] Jang E, Gu S and Poole B 2016 Categorical reparameterization with gumbel-softmax (arXiv:1611.01144)

[25] Maaløe L, Fraccaro M and Winther O 2017 Semi-supervised generation with cluster-aware generative models (arXiv:1704.00637)

[26] Rolfe J T 2016 Discrete variational autoencoders (arXiv:1609.02200)

[27] Vahdat A, Macready W, Bian Z, Khoshaman A and Andriyash E 2018 DVAE++: discrete variational autoencoders with overlapping transformations *Int. Conf. on Machine Learning* (*PMLR*) pp 5035–44

[28] Vahdat A, Andriyash E and Macready W 2018 *Advances in Neural Information Processing Systems* vol 31

[29] Khoshaman A, Vinci W, Denis B, Andriyash E, Sadeghi H and Amin M H 2019 *Quantum Sci. Technol.* **4** 014001

[30] Khoshaman A H and Amin M 2018 *Advances in Neural Information Processing Systems* vol 31

[31] Vinci W, Buffoni L, Sadeghi H, Khoshaman A, Andriyash E and Amin M 2020 *Mach. Learn.: Sci. Technol.* **1** 045028

[32] Vahdat A, Andriyash E and Macready W 2020 Undirected graphical models as approximate posteriors *Int. Conf. on Machine Learning* (PMLR) pp 9680–9

[33] An J and Cho S 2015 *Spec. Lecture IE* **2** 1–18

[34] Xu H *et al* 2018 Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications *Proc. 2018 World Wide Web Conf.* pp 187–96

[35] Chen R Q, Shi G H, Zhao W L and Liang C H 2019 A joint model for it operation series prediction and anomaly detection (arXiv:1910.03818)

[36] Wang X, Du Y, Lin S, Cui P, Shen Y and Yang Y 2020 *Knowl.-Based Syst.* **190** 105187

[37] Zhang C and Chen Y 2019 VELC: a new variational autoencoder based model for time series anomaly detection 70 (arXiv:1907.01702)

[38] Zhang C, Song D, Chen Y, Feng X, Lumezanu C, Cheng W, Ni J, Zong B, Chen H and Chawla N V 2019 A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data *Proc. AAAI Conf. on Artificial Intelligence* vol 33 pp 1409–16

[39] Park D, Hoshi Y and Kemp C C 2018 *IEEE Robot. Autom. Lett.* **3** 1544–51

[40] Su Y, Zhao Y, Niu C, Liu R, Sun W and Pei D 2019 Robust anomaly detection for multivariate time series through stochastic recurrent neural network *Proc. 25th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining* pp 2828–37

[41] Memarzadeh M, Matthews B and Avrekh I 2020 *Aerospace* **7** 115

[42] Yang S, Lee S and Yee K 2022 *Eng. Comput.* **39** 2239–55

[43] Kang Y E, Yang S and Yee K 2022 *Phys. Fluids* **34** 076103

[44] Smolensky P 1986 Information processing in dynamical systems: foundations of harmony theory *Technical Report* Colorado University at Boulder Department of Computer Science

[45] Van Den Oord A *et al* 2017 *Advances in Neural Information Processing Systems* vol 30

[46] Bartler A, Wiewel F, Mauch L and Yang B 2019 Training variational autoencoders with discrete latent variables using importance sampling *27th European Signal Processing Conf.* (*EUSIPCO*) (IEEE) pp 1–5

[47] Fajtl J, Argyriou V, Monekosso D and Remagnino P 2020 Latent Bernoulli autoencoder *Int. Conf. on Machine Learning* (PMLR) pp 2964–74

[48] Welling M, Rosen-Zvi M and Hinton G E 2004 *Advances in Neural Information Processing Systems* vol 17

[49] Salakhutdinov R, Mnih A and Hinton G 2007 Restricted Boltzmann machines for collaborative filtering *Proc. 24th Int. Conf. on Machine Learning* pp 791–8

[50] Hinton G E 2012 A practical guide to training restricted Boltzmann machines *Neural Networks: Tricks of the Trade* ed G Montavon, G B Orr and K R Müller (Springer) pp 599–619

[51] Fischer A and Igel C 2014 *Pattern Recognit.* **47** 25–39

[52] Hinton G E 2002 *Neural Comput.* **14** 1771–800

[53] Tieleman T 2008 Training restricted Boltzmann machines using approximations to the likelihood gradient *Proc. 25th Int. Conf. on Machine Learning* pp 1064–71

[54] Tieleman T and Hinton G 2009 Using fast weights to improve persistent contrastive divergence *Proc. 26th Annual Int. Conf. on Machine Learning* pp 1033–40

[55] Kingma D P and Welling M 2019 *Found. Trends® Mach. Learn.* **12** 307–92

[56] Mnih A and Gregor K 2014 Neural variational inference and learning in belief networks *Int. Conf. on Machine Learning* (PMLR) pp 1791–9

[57] Paisley J, Blei D and Jordan M 2012 Variational Bayesian inference with stochastic search (arXiv:1206.6430)

[58] Gu S, Levine S, Sutskever I and Mnih A 2015 MuProp: unbiased backpropagation for stochastic neural networks (arXiv:1511.05176)

[59] Bengio Y, Léonard N and Courville A 2013 Estimating or propagating gradients through stochastic neurons for conditional computation (arXiv:1308.3432)

[60] Maddison C J, Mnih A and Teh Y W 2016 The concrete distribution: a continuous relaxation of discrete random variables (arXiv:1611.00712)

[61] Amin M H, Andriyash E, Rolfe J, Kulchytskyy B and Melko R 2018 *Phys. Rev.* X **8** 021050

[62] Paszke A *et al* 2019 *Advances in Neural Information Processing Systems* vol 32

[63] Kingma D P and Ba J 2014 Adam: a method for stochastic optimization (arXiv:1412.6980)

[64] Li Y, Chen Z, Zha D, Zhou K, Jin H, Chen H and Hu X 2021 AutoOD: neural architecture search for outlier detection *37th Int. Conf. on Data Engineering* (*ICDE*) (IEEE) pp 2117–22

[65] Du Y and Mordatch I 2019 *Advances in Neural Information Processing Systems* vol 32

[66] Sønderby C K, Raiko T, Maaløe L, Sønderby S K and Winther O 2016 *Advances in Neural Information Processing Systems* vol 29

[67] Loaiza-Ganem G and Cunningham J P 2019 *Advances in Neural Information Processing Systems* vol 32

[68] Soenen J, Van Wolputte E, Perini L, Vercruyssen V, Meert W, Davis J and Blockeel H 2021 The effect of hyperparameter tuning on the comparative evaluation of unsupervised anomaly detection methods *Proc. KDD'21 Workshop on Outlier Detection and Description* (*Outlier Detection and Description Organising Committee*) pp 1–9

[69] Antoniadis I, Vercruyssen V and Davis J 2022 Systematic evaluation of CASH search strategies for unsupervised anomaly detection *Fourth Int. Workshop on Learning With Imbalanced Domains: Theory and Applications* (PMLR) pp 8–22

[70] Federal Aviation Administration 2004 Advisory circular 120–82 (www.faa.gov/documentLibrary/media/Advisory_Circular/AC_120-82.pdf)

[71] Dillman B G, Wilt D, Pruchnicki S, Ball M and Pomeroy M 2015 Flight operational quality assurance (FOQA)–do exceedances tell the story? *18th Int. Symp. on Aviation Psychology* p 354

[72] Benedetti M, Garcia-Pintos D, Perdomo O, Leyton-Ortega V, Nam Y and Perdomo-Ortiz A 2019 *npj Quantum Inf.* **5** 1–9

[73] Salakhutdinov R R 2009 *Advances in Neural Information Processing Systems* vol 22

[74] Desjardins G, Courville A, Bengio Y, Vincent P and Delalleau O 2010 Parallel tempering for training of restricted Boltzmann machines *Proc. 13th Int. Conf. on Artificial Intelligence and Statistics* (PMLR) pp 145–52

[75] Cho K, Raiko T and Ilin A 2010 Parallel tempering is efficient for learning restricted Boltzmann machines *The 2010 Int. Joint Conf. on Neural Networks* (*IJCNN*) (IEEE) pp 1–8

[76] Tanaka A and Tomiya A 2017 Towards reduction of autocorrelation in HMC by machine learning (arXiv:1712.03893)

[77] Matthews Bryan and Oza Nikunj Discovery in Aeronautics Systems Health (DASHlink) website 2018 Sample commercial-flight data from DASHlink project *Aviation Safety Program in NASA's Aeronautics Research Mission Directorate* (available at: https://c3.ndc.nasa.gov/dashlink/projects/85/)

[78] Matthews Bryan Discovery in Aeronautics Systems Health (DASHlink) website 2022 Curated 4 class anomaly detection data set *Aviation Safety Program in NASA's Aeronautics Research Mission Directorate* (available at: https://c3.ndc.nasa.gov/dashlink/resources/1018/)

[79] Fu M C 2006 Gradient estimation *Handbooks in Operations Research and Management Science* ed S G Henderson and B L Nelson (Elsevier) pp 575–616

[80] Williams R J 1992 *Mach. Learn.* **8** 229–56

[81] Glynn P W 1990 *Commun. ACM* **33** 75–84

[82] Grathwohl W, Choi D, Wu Y, Roeder G and Duvenaud D 2017 Backpropagation through the void: optimizing control variates for black-box gradient estimation (arXiv:1711.00123)

[83] Kuhn H and Tucker A 1951 Nonlinear programming *Proc. 2nd Berkeley Symp.* pp 481–92

[84] Karush W 1939 Minima of functions of several variables with inequalities as side constraints *MSc Thesis* Department of Mathematics, University of Chicago

[85] Song Y and Kingma D P 2021 How to train your energy-based models (arXiv:2101.03288)