
Analytical Probability Distributions and EM-Learning for Deep Generative Networks

Randall Balestriero
ECE Department
Rice University

Sébastien Paris
Aix-Marseille Univ, Université de Toulon,
CNRS, LIS, Toulon, France

Richard G. Baraniuk
ECE Department
Rice University

Abstract

Deep Generative Networks (DGNs) with probabilistic modeling of their output and latent space are currently trained via Variational Autoencoders (VAEs). In the absence of a known analytical form for the posterior and likelihood expectation, VAEs resort to approximations, including (Amortized) Variational Inference (AVI) and Monte-Carlo (MC) sampling. We exploit the Continuous Piecewise Affine (CPA) property of modern DGNs to derive their posterior and marginal distributions as well as the latter’s first moments. These findings enable us to derive an analytical Expectation-Maximization (EM) algorithm that enables gradient-free DGN learning. We demonstrate empirically that EM training of DGNs produces greater likelihood than VAE training. Our findings will guide the design of new VAE AVI that better approximate the true posterior and open avenues to apply standard statistical tools for model comparison, anomaly detection, and missing data imputation.

1 Introduction

Deep Generative Networks (DGNs), which map a low-dimensional latent variable z to a higher-dimensional generated sample x are the state-of-the-art methods for a range of machine learning applications, including anomaly detection, data generation, likelihood estimation, and exploratory analysis across a wide variety of datasets [11, 32, 40, 39].

Training of DGNs roughly falls into two camps: (i) by leveraging an adversarial network as in a Generative Adversarial Network (GAN) [23] to turn the method into an adversarial game; and (ii) by modeling the latent variable and observed variables as random variables and performing some flavor of likelihood maximization training. A widely used solution to likelihood based DGN training is via Variational Autoencoders (VAEs) [34]. The popularity of the VAE is due to its intuitive and interpretable loss function, which is obtained from likelihood estimation, and its ability to exploit standard estimation techniques ported from the probabilistic graphical models literature.

Yet, VAEs only offer an *approximate* solution for likelihood based training of DGNs. In fact, all current VAEs employ three major approximation steps in the likelihood maximization process. First, the true (unknown) posterior is approximated by a variational distribution. This estimate is governed by some free parameters that must be optimized to fit the variational distribution to the true posterior. VAEs estimate such parameters by means of an alternative network, the *encoder*, with the datum as input and the predicted optimal parameters as output. This step is referred to as Amortized Variational Inference (AVI), as it removes the explicit, per datum, optimization by a single deep network (DN) pass. Second, as in any latent variable model, the complete likelihood is estimated by a lower bound (ELBO) obtained from the expectation of the likelihood taken under the posterior or variational distribution. With a DGN, this expectation is unknown and thus VAEs estimate the ELBO by Monte-Carlo (MC) sampling. Third, the maximization of the MC estimated ELBO, which drives

the parameters of the encoder to better model the data distribution and the encoder to produce better variational parameter estimates, is performed by some flavor of gradient descend (GD).

These VAE approximation steps enable rapid training and test-time inference of DGNs. However, due to the lack of analytical forms for the posterior, ELBO, and explicit (gradient free) parameter updates, it is not possible to measure the above steps' quality or effectively improve them. Since the true posterior and expectation are unknown, current VAE research roughly fall into three camps: (i) on developing new and more complex output and latent distributions [45, 38] such as the truncated distribution; (ii) on improving the various estimation steps by introducing complex MC sampling with importance re-weighted sampling [13]; on providing different estimates of the posterior with moment matching techniques [17, 31].

In this paper, we advance both the theory and practice of VAEs by computing the exact analytical posterior and marginal distributions of any DGN employing continuous piecewise affine (CPA) nonlinearities. The knowledge of these distributions enables us to perform exact inference without resorting to AVI or MC-sampling and to train the DGN in a gradient-free manner with guaranteed convergence.

The analytical distributions we obtain provide first-of-their-kind insights into (i) how DGNs model the data distributions à la Mixture of Probabilistic Principal Component Analysis (MPPCA), (ii) how inference is performed and is akin Generative Latent Optimization models [12], (iii) the roles of each DGN parameter and how are they updated, and (iv) the impact of DGN architecture and regularization choice in the form of the DGN distributions and layer weights. The exact likelihood and marginal computation also enables the use of standard statistical model comparison tools such as the Akkaike Information Criterion (AIC) [1] and Bayesian Information Criterion (BIC) [49] and inspires new reliable anomaly detection approaches.

Having the exact posterior also enables us to quantify the approximation error of the AVI and MC sampling of VAEs and guide the development of VAEs by leveraging the analytical posterior to design more adapted variational distributions. In fact, current VAEs suffer from occasional training instabilities [60, 37]; we validate the empirical observation that VAEs training instabilities emerge from an inadequate variational estimation of the posterior.

We summarize our main contributions as follows:

[C1] We leverage the CPA property of current DGNs to obtain the analytical form of their conditional, marginal, and posterior distributions, which are mixtures of truncated Gaussians and relate DGN density modeling to MPPCA and MFA (Sec. 3.1). We develop new algorithms and methods to compute the DGN latent space partition, per-region affine mappings, and per-region Gaussian integration (Sec. 3.2).

[C2] We leverage the analytical form of a DGN's posterior distribution to obtain its first two moments. We then leverage these moments to obtain the analytical expectation of the complete likelihood with respect to the DGN posterior (E-step), which enables *encoder-free EM training with guaranteed convergence* (Sec. 4.1). We also derive the analytical M-step, which enables for the first time *guaranteed and rapid gradient-free learning of DGNs* (Sec. 4.2). The analytical E-step allows to interpret how the expected latent representation of an input is formed while the M-step demonstrates how information is propagated through layers akin backpropagation encounter in gradient descent.

[C3] We compare our exact E-step to standard VAE training to demonstrate that the VAE inference step is to blame for unstable training. We also demonstrate how EM-based DGN training provides much faster and stable convergence (Sec. 4.3); and provide new directions to leverage the analytical distributions to improve VAE models.

Reproducible code for all experiments and figures will be provided on Github at <https://github.com/RandallBalestriero/EMDGN.git>. The proofs of all results are provided in the Supplementary Material.

2 Background

Max-Affine Spline Deep Generative Networks. A deep generative network (DGN) is an operator g that maps a (typically low-dimensional) latent vector $\mathbf{z} \in \mathbb{R}^S$ to an observation $\mathbf{x} \in \mathbb{R}^D$ ¹ by composing L intermediate *layer* mappings g^ℓ , $\ell = 1, \dots, L$, that combine affine operators such as the *fully connected operator* (simply an affine transformation defined by weight matrix \mathbf{W}^ℓ and bias vector \mathbf{v}^ℓ), *convolution operator* (with circulant \mathbf{W}^ℓ), and, nonlinear operators such as the *activation operator* (applying a scalar nonlinearity such as the ubiquitous ReLU), or the *(max-)upsampling operator*; definitions of these operators can be found in [22].

In this paper, we focus on DGNs employing arbitrary affine operators and continuous piecewise affine (CPA) nonlinearities, such as the ReLU, leaky-ReLU, and absolute value activations, and spatial/channel max-pooling. In this case, the entire DGN is the composition of *Max-Affine Spline Operators* (MASOs) [5] and is overall a CPA operator [58, 2, 47, 55]. As such, DGNs inherit a latent space partition Ω and a per-region affine mapping

$$g(\mathbf{z}) = \mathbf{A}_\omega \mathbf{z} + \mathbf{b}_\omega, \forall \omega \in \Omega, \quad (1)$$

where the per-region slope and bias parameters are a function of the per-layer parameters $\mathbf{W}_\ell, \mathbf{b}_\ell$. For various properties of such CPA DGNs, see [7] and for details on the partition, see [6]. In this paper we will make explicit the per-region affine mappings; to this end, it is practical to encode the derivatives of the DGN nonlinearities in the matrices \mathbf{D}_ℓ . For activation operators, this is a square diagonal matrix with values $\in \{\eta, 1\}$ ($\eta > 0$ for leaky-ReLU, $\eta = 0$ for ReLU, and $\eta = -1$ for absolute value). For the max-pooling operator, it is a rectangular matrix filled with $\{0, 1\}$ values based on the pooling $\arg \max$. We thus obtain

$$\mathbf{A}_\omega = \mathbf{W}^L \mathbf{D}_\omega^{L-1} \mathbf{W}^{L-1} \dots \mathbf{D}_\omega^1 \mathbf{W}^1 \quad \text{and} \quad \mathbf{b}_\omega = \mathbf{v}^L + \sum_{i=1}^{L-1} \mathbf{W}^L \mathbf{D}_\omega^{L-1} \mathbf{W}^{L-1} \dots \mathbf{D}_\omega^i \mathbf{v}^i. \quad (2)$$

Throughout the rest of the paper, the upper index will indicate the layer and not a power.

Variational Expectation-Maximization. A Probabilistic Graphical Model (PGM) combines probability and graph theory into an organized data structure that expresses the relationships between a collection of random variables: the *observed* variables collected into \mathbf{x} and the *latent*, or unobserved, variables collected into \mathbf{z} [33]. The parameters θ that govern the PGM probability distributions are learned from observations $\mathbf{x}_i \sim \mathbf{x}, i = 1, \dots, N$, requiring estimation of the unobserved $\mathbf{z}_i, \forall i$. This inference-optimization is commonly done with the Expectation-Maximization (EM) algorithm [16].

The EM algorithm consists of (i) estimating each \mathbf{z}_i from the Expectation of the complete log-density taken with respect to the posterior distribution under the current parameters at time t ; (ii) Maximizing the estimated complete log-likelihood to produce the updated parameters θ_{t+1} . The estimated complete log-likelihood obtained from the E-step is a tight lower bound to the true complete log-likelihood; this lower bound is maximized in the M-step. This process has many attractive theoretical properties, including guaranteed convergence to a local minimum of the likelihood [35].

In the absence of closed form or tractable posterior, an alternative (non-tight) lower bound can be obtained by using a *variational distribution* instead. This distribution is governed by parameters γ that are optimized to make this distribution as close as possible to the true posterior. This process results in a *variational E* (VE) step [3] or *variational inference* (VI). The *tightness* of the lower bound is measured by the KL-divergence between the variational and true posterior distributions. Minimization of this divergence cannot be done directly (due to the absence of tractable posterior) but rather indirectly by maximizing the so-called evidence lower bound (ELBO) via

$$\log(p(\mathbf{x})) = \underbrace{\mathbb{E}_{q(\mathbf{z}|\gamma)}[\log(p(\mathbf{x}, \mathbf{z}|\theta))]}_{\text{ELBO}} + H(q(\mathbf{z}|\gamma)) + \text{KL}(q(\mathbf{z}|\gamma) || p(\mathbf{z}|\mathbf{x}, \theta)), \quad (3)$$

with q the variational distribution and H the (differential) entropy. Maximization the ELBO with respect to γ produces the γ^* that adapts $q(\mathbf{z}|\gamma^*)$ to fit as closely as possible to the true posterior. Finally, maximizing the ELBO with respect to the PGM parameters θ provides θ_{t+1} ; this can be performed on the entire dataset or on mini-batches [29].

¹Note that we do not require that $S < D$.

Variational AutoEncoders. A *Variational AutoEncoder* (VAE) uses a minimal *probabilistic graphical model* (PGM) with just a few nodes but highly nonlinear inter-node relations [36, 56]. The use of DNs to model the nonlinear relations originated in [46, 21, 42] and has been born again with VAEs [34]. Many variants have been developed but the core model consists of modeling the latent distribution over z with a Gaussian or uniform distribution and then modeling the data distribution as $x = g(z) + \epsilon$ with ϵ some noise distribution and g a DGN. Learning the DGN/PGM parameters requires inference of the latent variables z . This inference is done in VAEs by producing an *amortized* VI where a second *encoder* DN f produces $\gamma_n^* = f(x_n)$ from (3). Hence, the encoder is fed with an observation x and outputs its estimate of the optimal variational parameters that minimizes the KL-divergence between the variational distribution and true posterior. During learning, the encoder adapts to make better estimates $f(x_n)$ of the optimum parameters γ_n . Then, the ELBO is estimated with some flavor of Monte-Carlo (MC) sampling (since its analytical form is not known) and the maximization of the θ parameters is solved iteratively using some flavor of gradient descent.

3 Posterior and Marginal Distributions of Deep Generative Networks

We now derive analytical forms of the key DGN distributions by exploiting the CPA property. In Sec. 4 we will use this result to derive the EM learning algorithm for DGNs and study the VAE inference approximation versus the analytical one.

Our key insight is that a CPA DGN consists of an implicit latent space partition and an associated per-region affine mapping (recall (1)). In a DGN, propagating a latent datum z through the layers progressively builds the A_ω, b_ω . We now demonstrate that turning this region selection process explicit, the analytical DGN marginal and posterior distributions can be obtained.

3.1 Conditional, Marginal and Posterior Distributions of Deep Generative Networks

Throughout the paper we will consider the commonly employed case of a centered Gaussian latent prior and centered Gaussian noise [59] as

$$p(x|z) = \phi(x; g(z), \Sigma_x), \quad p(z) = \phi(z; 0, \Sigma_z), \quad (4)$$

with ϕ the multivariate Gaussian density function with given mean and covariance matrix [14]. When using CPA DGNs, the generator mapping is continuous and piecewise affine with an underlying latent space partition, and per-region mapping as given by (1). We can thus obtain the analytical form of the conditional distribution of x given the latent vector z as follows.

Lemma 1. *The DGN conditional distribution is given by $p(x|z) = \sum_{\omega \in \Omega} \mathbb{1}_{z \in \omega} \phi(x; A_\omega z + b_\omega, \Sigma_x)$ with per-region parameters from (2).*

This type of data modeling is closely related to MPPCA [53] that combines multiple PPCAs [54] and MFA [20, 27] that combines multiple factor analyzers [25]. The associated PGMs represent the data distribution with R components and leverage an explicit categorical distribution $t \sim \text{Cat}(\pi)$, leading to the conditional input distributions $x|(z, t) = \sum_{r=1}^R \mathbb{1}_{t=r} (W_r z + v_r) + \epsilon$, with W_r, v_r denoting the per-component affine parameters and with Σ_x diagonal (MPPCA) or fully occupied (MFA) and $z \sim \mathcal{N}(\mu_z, \Sigma_z)$. Note, however, that neither MPPCA nor MFA impose continuity in the $(t, z) \mapsto x$ mapping as opposed to a DGN. To formalize this, consider an (arbitrary) ordering of the DGN latent space regions as $\omega_1, \dots, \omega_R$ with $R = \text{Card}(\Omega)$; we also denote by Φ_ω the cumulative density function on ω_r (integral of the density function on ω_r).

Proposition 1. *A DGN with distributions given by (4) corresponds to a continuous MPPCA (or MFA) model with implicit categorical variable given by $p(t = r) = \Phi_{\omega_r}(0, \Sigma_z)$, $W_r = A_{\omega_r}$, $v_r = b_{\omega_r}$, $R = \text{Card}(\Omega)$ and $\Sigma_x = \sigma I$ (or full Σ_x).*

Note that this result generalizes the result of [41] which related linear and shallow DGNs to PPCA, as in the linear regime one has $g(z) = Wz + b + \epsilon$. We now consider the marginal $p(x)$ and posterior $p(z|x)$ distributions. The former will be of use to compute the likelihood, while the latter will enable us to derive the analytical E-step in the next section.

Theorem 1. *The marginal and posterior distributions of a CPA DGN are given by*

$$p(x) = \sum_{\omega \in \Omega} \phi(x; b_\omega, \Sigma_x + A_\omega \Sigma_z A_\omega^T) \Phi_\omega(\mu_\omega(x), \Sigma_\omega), \quad (5)$$

$$p(\mathbf{z}|\mathbf{x}) = p(\mathbf{x})^{-1} \sum_{\omega \in \Omega} \mathbb{1}_{\mathbf{z} \in \omega} \phi(\mathbf{x}; \mathbf{b}_\omega, \Sigma_\omega + \mathbf{A}_\omega \Sigma_\mathbf{z} \mathbf{A}_\omega^T) \phi(\mathbf{z}; \boldsymbol{\mu}_\omega(\mathbf{x}), \Sigma_\omega), \quad (6)$$

with

$$\boldsymbol{\mu}_\omega(\mathbf{x}) = \Sigma_\omega (\mathbf{A}_\omega^T \Sigma_\mathbf{x}^{-1} (\mathbf{x} - \mathbf{b}_\omega)), \quad \Sigma_\omega = (\Sigma_\mathbf{z}^{-1} + \mathbf{A}_\omega^T \Sigma_\mathbf{x}^{-1} \mathbf{A}_\omega)^{-1}. \quad (7)$$

The distribution $\Phi_\omega(\boldsymbol{\mu}_\omega(\mathbf{x}), \Sigma_\omega)$ is derived in the next section. For both the marginal and the posterior distribution, on each partition region, there exists a mean $\boldsymbol{\mu}_\omega(\mathbf{x})$ and covariance Σ_ω that we can interpret. For that purpose, consider $\Sigma_\mathbf{x} = I$, $\Sigma_\mathbf{z} = I$ to obtain $\boldsymbol{\mu}_\omega(\mathbf{x}) = (I + \mathbf{A}_\omega^T \mathbf{A}_\omega)^{-1} \mathbf{A}_\omega^T (\mathbf{x} - \mathbf{b}_\omega)$. That is, the bias of the per-region affine mapping is removed from the input which is then mapped back to the latent space via \mathbf{A}_ω^T and whitened by the “regularized” inverse of the correlation matrix of \mathbf{A}_ω . Note that \mathbf{A}_ω^T backpropagates the signal from the output to the latent space in the same way that gradients are backpropagated during gradient learning of a DN. We further highlight the specific form of the posterior as being a mixture model of truncated Gaussians [30], a truncated Gaussian being a Gaussian distribution for which the domain \mathbb{R}^S has been constrained to a (convex) sub-domain, ω in our case.

Proposition 2. *The DGN posterior distribution is a mixture of $\text{Card}(\Omega)$ truncated Gaussians, each truncated on a different polytope $\omega \in \Omega$ with mean $\boldsymbol{\mu}_\omega(\mathbf{x})$ and covariance Σ_ω from (7).*

Zero-Noise Limit and Generative Latent Optimization (GLO) Models. In the zero-noise limit ($\Sigma_\mathbf{x} = \sigma I$ and $\sigma \rightarrow 0$) the posterior takes a very special form. Denote by $\mathbf{z}^*(\mathbf{x}) \triangleq \arg \min_{\mathbf{z}} \|\mathbf{x} - g(\mathbf{z})\|_2^2 + \mathbf{z}^T \Sigma_\mathbf{z} \mathbf{z}$, the (regularized) latent vector that produces the closest output from an observation \mathbf{x} .

Lemma 2. *In the zero-noise limit, the DGN posterior distribution converges to a Dirac positioned in the \mathbf{z} -space at $\mathbf{z}^*(\mathbf{x})$ as $\lim_{\sigma \rightarrow 0} p(\mathbf{z}|\mathbf{x}) = \delta(\mathbf{z} - \mathbf{z}^*(\mathbf{x}))$.*

Interestingly, GLO [12] performs DGN training by first inferring a latent vector akin to $\mathbf{z}^*(\mathbf{x})$ but without the ℓ_2 regularization $\mathbf{z}^T \Sigma_\mathbf{z} \mathbf{z}$, which is often replaced by a \mathbf{z} truncation/clipping.

Proposition 3. *The GLO-inferred DGN latent variable associated to an observation \mathbf{x} corresponds to the maximum a posteriori estimate of the zero-noise limit posterior distribution and with uninformative prior (large $\Sigma_\mathbf{z}$) or with uniform prior $\mathbf{z} \sim \mathcal{U}([a, b])$ when using $[a, b]$ clipping).*

3.2 Gaussian Integration on the Deep Generative Network Latent Partition

We now turn to the computation of the DGN marginal (5) and posterior (6) distributions for which we need to integrate over all of the regions $\omega \in \Omega$ in the latent space partition.

Obtaining the DGN Partition. Each region $\omega \in \Omega$ is a polytope that can be explicitly described via a system of inequalities involving the up-to-layer ℓ mappings

$$\mathbf{A}_\omega^{1 \rightarrow \ell} \triangleq \mathbf{W}^\ell \mathbf{D}_\omega^{\ell-1} \mathbf{W}^{\ell-1} \dots \mathbf{D}_\omega^1 \mathbf{W}^1 \quad \text{and} \quad \mathbf{b}_\omega^{1 \rightarrow \ell} \triangleq \mathbf{v}^\ell + \sum_{i=1}^{\ell-1} \mathbf{W}^\ell \mathbf{D}_\omega^{\ell-1} \mathbf{W}^{\ell-1} \dots \mathbf{D}_\omega^i \mathbf{v}^i, \quad (8)$$

producing the pre-activation feature maps $\mathbf{h}^\ell(\mathbf{z}) \in \mathbb{R}^{D^\ell}$ by $\mathbf{h}^\ell(\mathbf{z}) = \mathbf{A}_\omega^{1 \rightarrow \ell} \mathbf{z} + \mathbf{b}_\omega^{1 \rightarrow \ell}$ and with $\mathbf{A}_\omega^{1 \rightarrow \ell} \in \mathbb{R}^{D^\ell \times S}$ and $\mathbf{b}_\omega^{1 \rightarrow \ell} \in \mathbb{R}^{D^\ell}$. Note that we have, in particular, that $\mathbf{A}_\omega^L = \mathbf{A}_\omega$ and $\mathbf{b}_\omega^L = \mathbf{b}_\omega$ from (2). When using standard activation functions such as (leaky-)ReLU or absolute value, the sign of the pre-activation defines the activation state; denote this by $\mathbf{q}^\ell = \text{sign}(\mathbf{h}^\ell(\mathbf{z}))$ and collect all of the per-layer signs into \mathbf{q} . Without degenerate weights, the sign patterns produced by $\mathbf{q}(\mathbf{z})$, $\forall \mathbf{z}$ are tied to the regions $\omega \in \Omega$; we will thus use interchangeably $\mathbf{q}(\mathbf{z})$ with $\mathbf{z} \in \omega$ and $\mathbf{q}(\omega)$.

Lemma 3. *The operator $\mathbf{z} \mapsto [\mathbf{q}^1(\mathbf{z}), \dots, \mathbf{q}^{L-1}(\mathbf{z})]$ is piecewise constant with a bijection between its image and Ω .*

Corollary 1. *The polyhedral region ω is given by*

$$\omega = \bigcap_{\ell=1}^{L-1} \{ \mathbf{z} \in \mathbb{R}^S : \mathbf{A}_\omega^{1 \rightarrow \ell} \mathbf{z} < -\mathbf{q}^\ell(\omega) \odot \mathbf{b}_\omega^{1 \rightarrow \ell} \},$$

with \odot the Hadamard product.

The above result tells us that the pre-activation signs locate on which side of each hyperplane the region ω is located, which provides a direct way to compute the \mathcal{H} -representation of ω from $\mathbf{q}(\mathbf{z})$

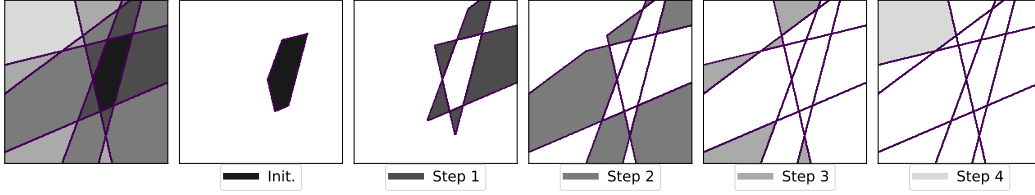


Figure 1: Recursive partition discovery for a DGN with $S = 2$ and $L = 2$, starting with an initial region obtained from a sampled latent vector \mathbf{z} (init). By walking on the faces of this region, neighboring regions sharing a common face are discovered (Step 1). Recursively repeating this process until no new region is discovered (Steps 2–4) provides the DGN latent space partition at left.

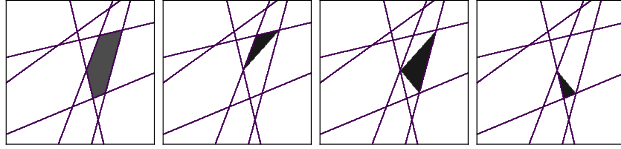


Figure 2: Triangulation $T(\omega)$ as per (9) of a polytopal region ω (left plot) obtained from the Delaunay Triangulation of the region vertices leading to 3 simplices (three right plots).

with $\mathbf{z} \in \omega$. To obtain the entire partition Ω , we propose a recursive scheme that starts from an initial region (or sample \mathbf{z}) and walks on its faces to discover the neighboring regions. This process is repeated on the newly discovered regions until no new region is discovered. We detail this exploration procedure in Appendix A and illustrate it in Fig. 1.

Gaussian Integration on ω . The Gaussian integral on a region ω (and its moments) cannot in general be obtained by direct integration unless ω is a rectangular region [51, 9] or is polytopal with at most S faces [52]. In general, the DGN regions $\omega \in \Omega$ will have at least $S + 1$ faces, as they are closed polytopes in \mathbb{R}^S . To leverage the known integral forms, we propose to first decompose a DGN region ω into simplices ($S + 1$ -face polytopes in our case [44]) and then further decompose each simplex into open polytopes with at most S faces, which enables the use of [52]. In our case, we perform the simplex decomposition with the Delaunay triangulation [15] denoted as $T(\omega)$ with

$$T(\omega) \triangleq \{\Delta_1, \dots, \Delta_{\text{Card}(T(\omega))}\}, \text{ with } \bigcup_{i=1}^{\text{Card}(T(\omega))} \Delta_i = \omega \text{ and } \Delta_i \cap \Delta_j = \emptyset, \forall i \neq j, \quad (9)$$

where each Δ_i is a simplex defined by the half-spaces $\Delta_i = \bigcap_{s=1}^{S+1} H_{i,s}$. This process is illustrated in Fig. 2. The decomposition of each simplex into open polytopes with less than $S + 1$ faces is performed by employing the standard inclusion-exclusion principle [10] leading to the following result.

Lemma 4. *The integral of any integrable function g on a polytopal region $\omega \in \Omega$ can be decomposed into integration over open polytopes of at most S faces via*

$$\int_{\omega} g(\mathbf{z}) d\mathbf{z} = \sum_{\Delta \in T(\omega)} \sum_{(s, V) \in H(\Delta)} s \int_V g(\mathbf{z}) d\mathbf{z}.$$

with $H(\Delta_i) \triangleq \{((-1)^{|J|+S}, \bigcap_{j \in J} H_{i,j}), J \subseteq \{1, \dots, S+1\}, |J| \leq S\}$.

From the above result, we can apply the known form of the Gaussian integral on a polytopal region with fewer than S faces and obtain the form of the integral and moments as provided in Appendix B, where detailed pseudo code is provided.

Remark: The integral performed as per Lemma 4 is computationally expensive, particularly with respect to the latent space dimension S . This is the current main practical limitation of performing the analytical computation of the DGN posterior (and thus the E-step). A more elaborated discussion plus several solutions are provided in the next section; see also Appendix I for the asymptotic computational complexity details.

Visualization of the Marginal and Posterior Distributions. To illustrate our theoretical development so far, we now visualize the posterior and marginal distributions of a randomly initialized DGN in a low-dimensional space $D = 2$ and with latent dimension $S = 1$. (See Appendix J for the architectural details of the DGN.) We depict the obtained distributions as well as the generated samples in Fig. 3. We also plot the posterior distribution based on one observation obtained via $g(\mathbf{z}_0)$ given a sampled \mathbf{z}_0 from the \mathbf{z} distribution and one noisy observation $g(\mathbf{z}_0) + \epsilon_0$ given a noise realization ϵ_0 .

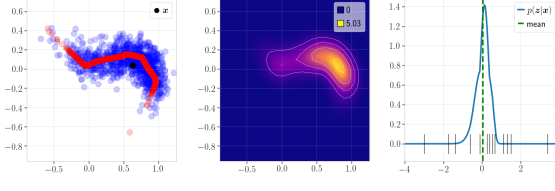


Figure 3: **Left:** Noiseless generated samples $g(\mathbf{z})$ in red and noisy samples $g(\mathbf{z}) + \epsilon$ in blue, with $\Sigma_{\mathbf{x}} = 0.1I$, $\Sigma_{\mathbf{z}} = I$. **Middle:** marginal distribution $p(\mathbf{x})$ from (5). **Right:** the posterior distribution $p(\mathbf{z}|\mathbf{x})$ from (6) (blue), its expectation (green) and the position of the region limits (black), with sample point \mathbf{x} depicted in black in the left figure.

4 Expectation-Maximization Learning of Deep Generative Networks

We now derive an analytical Expectation-Maximization training algorithm for CPA DGNs based on the results of the previous sections. We then compare DGN training via EM and AVI and leverage the exact complete likelihood to perform model selection and study the VAE approximation error.

4.1 Expectation Step

The E-step infers the latent (unobserved) variables associated to the generation of each observation \mathbf{x} by taking the expectation of the log of the complete likelihood with respect to the posterior distribution (6). We denote the per-region moments of the DGN posterior (from Appendix B) by $\mathbb{E}_{\mathbf{z}|\mathbf{x}}[\mathbb{1}_{\mathbf{z} \in \omega}] \triangleq e_{\omega}^0(\mathbf{x})$, $\mathbb{E}_{\mathbf{z}|\mathbf{x}}[\mathbf{z} \mathbb{1}_{\mathbf{z} \in \omega}] \triangleq \mathbf{e}_{\omega}^1(\mathbf{x})$ and $\mathbb{E}_{\mathbf{z}|\mathbf{x}}[\mathbf{z} \mathbf{z}^T \mathbb{1}_{\mathbf{z} \in \omega}] \triangleq \mathbf{E}_{\omega}^2(\mathbf{x})$; we also have $\mathbf{e}^1(\mathbf{x}) \triangleq \mathbb{E}_{\mathbf{z}|\mathbf{x}}[\mathbf{z}] = \sum_{\omega} \mathbf{e}_{\omega}^1(\mathbf{x})$ and likewise for the second moment. We obtain the following E-step (the detailed derivations are in Appendix G.1)

$$\begin{aligned} E_{\mathbf{z}|\mathbf{x}}[\log(p(\mathbf{x}|\mathbf{z})p(\mathbf{z}))] = & -\frac{1}{2} \log\left((2\pi)^{S+D} |\det(\Sigma_{\mathbf{x}})| |\det(\Sigma_{\mathbf{z}})|\right) - \frac{1}{2} \text{Tr}(\Sigma_{\mathbf{z}}^{-1} \mathbf{E}^2(\mathbf{x})) \\ & - \frac{1}{2} \left(\mathbf{x}^T \Sigma_{\mathbf{x}}^{-1} \mathbf{x} - 2 \mathbf{x}^T \Sigma_{\mathbf{x}}^{-1} \left(\sum_{\omega} \mathbf{A}_{\omega} \mathbf{e}_{\omega}^1(\mathbf{x}) + \mathbf{b}_{\omega} e_{\omega}^0(\mathbf{x}) \right) \right. \\ & \left. + \sum_{\omega} \left[\text{Tr}(\mathbf{A}_{\omega}^T \Sigma_{\mathbf{x}}^{-1} \mathbf{A}_{\omega} \mathbf{E}_{\omega}^2(\mathbf{x})) + (e_{\omega}^0 \mathbf{b}_{\omega} + 2 \mathbf{A}_{\omega} \mathbf{e}_{\omega}^1(\mathbf{x}))^T \Sigma_{\mathbf{x}}^{-1} \mathbf{b}_{\omega} \right] \right). \end{aligned}$$

Note that the (per-region) moments involved in the E-step, such as $\mathbf{e}_{\omega}^1(\mathbf{x})$, are taken with respect to the current parameters ($\theta = \{\Sigma_{\mathbf{x}}, \Sigma_{\mathbf{z}}, (\mathbf{W}^{\ell}, \mathbf{v}^{\ell})_{\ell=1}^L\}$). That is, if gradient based optimization is leveraged to maximize the ELBO, then no gradient should be propagated through them. We can see from the above formula that the contributions of each region's affine parameters are weighted based on the posterior for each datum \mathbf{x} . That is, for each input \mathbf{x} , the posterior combines all of the per-region affine parameters as opposed to current forms of learning that only leverage the parameters involved on the specific region activated by the DGN input \mathbf{z} .

4.2 Maximization Step

Given the E-step, maximizing the ELBO can be done via some flavor of gradient based optimization. However, thanks to the analytical E-step and the Gaussian form of the involve distributions, there exists analytical form of this maximisation process (M-step) leading to the analytical M-step for DGNs. The formulas for all of the DGN parameters are provided in Appendix G. We provide here the analytical form for the bias $\mathbf{v}^{\ell*}$, for which we introduce $\mathbf{r}_{\omega}^{\ell}(\mathbf{x})$ as the expected reconstruction error of the DGN as

$$\begin{aligned} \mathbf{r}_{\omega}^{\ell}(\mathbf{x}) & \triangleq \left(\mathbf{x} - \sum_{i \neq \ell} \mathbf{A}_{\omega}^{i+1 \rightarrow L} \mathbf{D}_{\omega}^i \mathbf{v}^i \right) e_{\omega}^0(\mathbf{x}) - \mathbf{A}_{\omega} \mathbf{e}_{\omega}^1(\mathbf{x}) \quad (\text{expected residual without } \mathbf{v}^{\ell}), \\ \mathbf{v}^{\ell*} & = \left(\sum_{\mathbf{x}} \sum_{\omega} \mathbf{D}_{\omega}^{\ell} \mathbf{A}_{\omega}^{L \rightarrow \ell+1} \Sigma_{\mathbf{x}}^{-1} \mathbf{A}_{\omega}^{\ell+1 \rightarrow L} \mathbf{D}_{\omega}^{\ell} \right)^{-1} \left(\sum_{\mathbf{x}} \sum_{\omega \in \Omega} \underbrace{\mathbf{D}_{\omega}^{\ell} \mathbf{A}_{\omega}^{L \rightarrow \ell+1} \Sigma_{\mathbf{x}}^{-1} \mathbf{r}_{\omega}^{\ell}(\mathbf{x})}_{\text{residual back-propagated to layer } \ell} \right). \end{aligned}$$

Some interesting observations can be made based on the analytical form of these updates. First, the bias update is based on the residual of the reconstruction error with a DGN whose bias has been

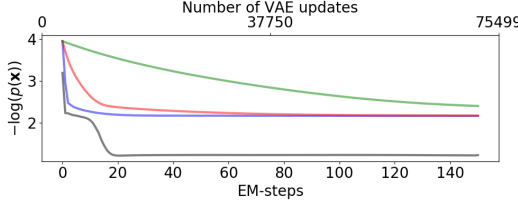


Figure 4: DGN training under EM (black) and VAE training with various learning rates for VAE (blue: 0.005, red: 0.001, green: 0.0001). In all cases, VAE converges to the maximum of its ELBO. The gap between the VAE and EM curves is due to the inability of the VAE’s AVI to correctly estimate the true posterior, pushing the VAE’s ELBO far from the true log-likelihood (recall (3)) and thus preventing it from precisely approximating the true data distribution.

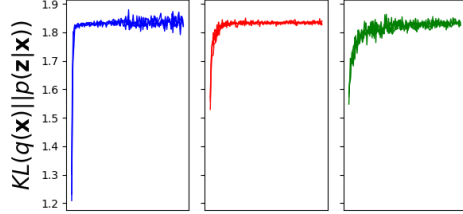


Figure 5: KL-divergence between a VAE variational distribution and the true DGN posterior when trained on a noisy circle dataset in 2D for 3 different learning rates. During learning, the DGN adapts such that $g(\mathbf{z}) + \epsilon$ models the data distribution based on the VAE’s estimated ELBO. As learning progresses, the true DGN posterior becomes harder to approximate by the VAE’s variational distribution in the AVI process. As such, even in this toy dataset the commonly employed Gaussian variational distribution is not rich enough to capture the multimodality of $p(\mathbf{z}|\mathbf{x})$ from (6).

removed; this residual is then backpropagated to the ℓ^{th} layer. The backpropagation is performed via the (transposed) backpropagation matrix as when performing gradient-based learning. Second, the updates of any parameter depend on each region parameter’s contribution based on the posterior moments and integrals, similarly to any mixture model. Third, all of the updates are whitened based on the backpropagation (or forward propagation) correlation matrix $\mathbf{A}_{\omega}^{\ell \rightarrow L}, \forall \omega, \forall \ell$. We study the impact of using a probabilistic priors on the layer weights such as Gaussian, Laplacian, or uniform which are related to the ℓ_2, ℓ_1 regularization, and weight clipping techniques in Appendix H.

4.3 Empirical Validation and VAE Comparison

We now numerically validate the above EM-steps on a simple problem involving data points on a radius 1 circle in 2D augmented with a Gaussian noise of standard deviation 0.05. We depict the EM-training of a 2-layer DGN with width of 8 with the VAE training. In all cases the DGNs are the same architecture, with same weight initialization, and the dataset is also identical between models with the same noise realizations. Thanks to the analytical form of the marginals, we can compute the true ELBO (without variational estimation of the true posterior) for the VAE during its training to monitor its ability to fit the data distribution. We depict this in Fig. 4. We observe how EM training converges faster to a lower negative log-likelihood. In addition, we see how all of the trained VAEs seem to converge to the same bound, which corresponds to the maximum of its ELBO, where the gap is induced by the use of a variational approximation of the true posterior. We further investigate this phenomenon in Fig. 5. In fact, we obtained from (6) and Prop. 2 that the posterior is a mixture of truncated Gaussian and the covariances are based on $\mathbf{A}_{\omega}^T \mathbf{A}_{\omega}$. Hence VAEs employing a single Gaussian as a variational distribution $q(\mathbf{x})$ will fail to correctly approximate the true posterior $p(\mathbf{z}|\mathbf{x})$, especially as training proceeds making the posterior covariance less diagonal. The absence of precise posterior estimation directly translated into VAE optimizing an ELBO objective being far from the tight log-likelihood bound, in turn hurting VAE ability to correctly approximate the data distribution as opposed to EM learning producing much smaller negative log-likelihood. Providing better posterior estimates is thus key to improve VAE performances. From this we conclude that multimodal variational distributions should be considered for VAEs regardless of the data at hand. We provide additional figures and experiments with various architectures in Appendix J.

5 Conclusions

We have derived the analytical form of the posterior, marginal, and conditional distributions of for CPA DGNs. This has enabled us to derive the EM-learning algorithm for DGNs that not only converges faster than state-of-the-art VAE training but also to a higher likelihood. Our analytical forms can be

leveraged to improve the variational distribution of VAEs, understand the form of analytical weight updates, study how a DGN infers the latent variable z from \mathbf{x} , and leverage standard statistical tools to perform model selection, anomaly detection and beyond.

In addition of these insights, our new EM learning algorithm can be used to train DGNs. The main computational bottlenecks are the width/depth of the DGN and the dimension of the latent space. One interesting research direction is developing approximations to our various analytical forms that enable fast inference/learning à la VAE but with better approximation and quantified approximation error. One promising direction would involve only computing and integrating the distributions on the regions $\omega \in \Omega$ where we know a priori that the posterior is nonzero.

Broader Impacts

We have derived the analytical form of the posterior, marginal and conditional distributions of Deep Generative Networks (DGNs) based on continuous piecewise affine architectures. Our approach provides an approximation-free alternative to VAEs to train DGNs. In addition to improving DGN algorithms, our analytical forms will enable researchers to probe more deeply into the inner workings of DGNs and VAE, making them more interpretable and thus trustworthy. Our calculations will also enable accurate anomaly detection and model selection, which should find wide application in sensitive applications where accurately computing the probability of a data point is crucial.

Acknowledgments

RB and RB were supported by NSF grants CCF-1911094, IIS-1838177, and IIS-1730574; ONR grants N00014-18-12571 and N00014-17-1-2551; AFOSR grant FA9550-18-1-0478; DARPA grant G001534-7500; and a Vannevar Bush Faculty Fellowship, ONR grant N00014-18-1-2047.

References

- [1] Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.
- [2] Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. *arXiv preprint arXiv:1611.01491*, 2016.
- [3] Hagai Attias. A variational bayesian framework for graphical models. In *Advances in neural information processing systems*, pages 209–215, 2000.
- [4] David Avis and Komei Fukuda. Reverse search for enumeration. *Discrete applied mathematics*, 65(1-3):21–46, 1996.
- [5] R. Balestrierio and R. G. Baraniuk. A spline theory of deep networks. In *Proc. Int. Conf. Mach. Learn.*, volume 80, pages 374–383, Jul. 2018.
- [6] Randall Balestrierio, Romain Cosentino, Behnaam Aazhang, and Richard Baraniuk. The geometry of deep networks: Power diagram subdivision. In *Advances in Neural Information Processing Systems 32*, pages 15806–15815. 2019.
- [7] Randall Balestrierio, Sebastien Paris, and Richard Baraniuk. Max-affine spline insights into deep generative networks. *arXiv preprint arXiv:2002.11912*, 2020.
- [8] C Bradford Barber, David P Dobkin, and Hannu Huuhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483, 1996.
- [9] Manjunath BG and Stefan Wilhelm. Moments calculation for the double truncated multivariate normal density. *Available at SSRN 1472153*, 2009.
- [10] Andreas Björklund, Thore Husfeldt, and Mikko Koivisto. Set partitioning via inclusion-exclusion. *SIAM Journal on Computing*, 39(2):546–563, 2009.
- [11] Merlijn Blaauw and Jordi Bonada. Modeling and transforming speech using variational autoencoders. *Morgan N, editor. Interspeech 2016; 2016 Sep 8-12; San Francisco, CA.[place unknown]: ISCA; 2016. p. 1770-4.*, 2016.
- [12] Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. Optimizing the latent space of generative networks. *arXiv preprint arXiv:1707.05776*, 2017.
- [13] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.
- [14] Morris H DeGroot and Mark J Schervish. *Probability and statistics*. Pearson Education, 2012.
- [15] Boris Delaunay et al. Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7(793-800):1–2, 1934.
- [16] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.
- [17] Adji B Dieng and John Paisley. Reweighted expectation maximization. *arXiv preprint arXiv:1906.05850*, 2019.

- [18] Martin E Dyer. The complexity of vertex enumeration methods. *Mathematics of Operations Research*, 8(3):381–402, 1983.
- [19] Thomas Gerstner and Markus Holtz. Algorithms for the cell enumeration and orthant decomposition of hyperplane arrangements. *University of Bonn*, 2006.
- [20] Zoubin Ghahramani, Geoffrey E Hinton, et al. The em algorithm for mixtures of factor analyzers. Technical report, Technical Report CRG-TR-96-1, University of Toronto, 1996.
- [21] Zoubin Ghahramani and Sam T Roweis. Learning nonlinear dynamical systems using an em algorithm. In *Advances in neural information processing systems*, pages 431–437, 1999.
- [22] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*, volume 1. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [23] I. J Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pages 2672–2680. MIT Press, 2014.
- [24] Branko Grünbaum. *Convex polytopes*, volume 221. Springer Science & Business Media, 2013.
- [25] Harry H Harman. *Modern factor analysis*. University of Chicago press, 1976.
- [26] David A Harville. Matrix algebra from a statistician’s perspective, 1998.
- [27] Geoffrey E Hinton, Peter Dayan, and Michael Revow. Modeling the manifolds of images of handwritten digits. *IEEE transactions on Neural Networks*, 8(1):65–74, 1997.
- [28] Francis Hirsch and Gilles Lacombe. *Elements of functional analysis*, volume 192. Springer Science & Business Media, 2012.
- [29] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.
- [30] William C Horrace. Some results on the multivariate truncated normal distribution. *Journal of multivariate analysis*, 94(1):209–221, 2005.
- [31] Chin-Wei Huang, Kris Sankaran, Eeshan Dhekane, Alexandre Lacoste, and Aaron Courville. Hierarchical importance weighted autoencoders. *arXiv preprint arXiv:1905.04866*, 2019.
- [32] Tadanobu Inoue, Subhjit Choudhury, Giovanni De Magistris, and Sakyasingha Dasgupta. Transfer learning from synthetic to real images using variational autoencoders for precise position detection. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2725–2729. IEEE, 2018.
- [33] Michael I Jordan. An introduction to probabilistic graphical models, 2003.
- [34] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [35] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
- [36] Harri Lappalainen and Antti Honkela. Bayesian non-linear independent component analysis by multi-layer perceptrons. In *Advances in independent component analysis*, pages 93–121. Springer, 2000.
- [37] Ruizhe Li, Xiao Li, Chenghua Lin, Matthew Collinson, and Rui Mao. A stable variational autoencoder for text modelling. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 594–599, Tokyo, Japan, October–November 2019. Association for Computational Linguistics.
- [38] Xiaopeng Li and James She. Collaborative variational autoencoder for recommender systems. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 305–314, 2017.
- [39] Jaechang Lim, Seongok Ryu, Jin Woo Kim, and Woo Youn Kim. Molecular generative model based on conditional variational autoencoder for de novo molecular design. *Journal of cheminformatics*, 10(1):1–9, 2018.
- [40] Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander Gaunt. Constrained graph variational autoencoders for molecule design. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7795–7804. Curran Associates, Inc., 2018.
- [41] James Lucas, George Tucker, Roger Grosse, and Mohammad Norouzi. Understanding posterior collapse in generative latent variable models. 2019.

- [42] David JC MacKay and Mark N Gibbs. Density networks. *Statistics and neural networks: advances at the interface*. Oxford University Press, Oxford, pages 129–144, 1999.
- [43] Theodore S Motzkin, Howard Raiffa, Gerald L Thompson, and Robert M Thrall. The double description method. *Contributions to the Theory of Games*, 2(28):51–73, 1953.
- [44] James R Munkres. *Elements of algebraic topology*. CRC Press, 2018.
- [45] Eric Nalisnick and Padhraic Smyth. Stick-breaking variational autoencoders. *arXiv preprint arXiv:1605.06197*, 2016.
- [46] Jong-Hoon Oh and H Sebastian Seung. Learning generative models with the up propagation algorithm. In *Advances in Neural Information Processing Systems*, pages 605–611, 1998.
- [47] Blaine Rister and Daniel L Rubin. Piecewise convexity of artificial neural networks. *Neural Networks*, 94:34–45, 2017.
- [48] Sam Roweis and Zoubin Ghahramani. A unifying review of linear gaussian models. *Neural computation*, 11(2):305–345, 1999.
- [49] Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.
- [50] Nora H Sleumer. Output-sensitive cell enumeration in hyperplane arrangements. *Nordic journal of computing*, 6(2):137–147, 1999.
- [51] Georges M Tallis. The moment generating function of the truncated multi-normal distribution. *Journal of the Royal Statistical Society: Series B (Methodological)*, 23(1):223–229, 1961.
- [52] GM Tallis. Plane truncation in normal populations. *Journal of the Royal Statistical Society: Series B (Methodological)*, 27(2):301–307, 1965.
- [53] Michael E Tipping and Christopher M Bishop. Mixtures of probabilistic principal component analyzers. *Neural computation*, 11(2):443–482, 1999.
- [54] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.
- [55] Michael Unser. A representer theorem for deep neural networks. *Journal of Machine Learning Research*, 20(110):1–30, 2019.
- [56] Harri Valpola. *Unsupervised learning of nonlinear dynamic state-space models*. Helsinki University of Technology, 2000.
- [57] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, pages 1–12, 2020.
- [58] S. Wang, X. Huang, and K. M. Junaid. Configuration of continuous piecewise-linear neural networks. *IEEE Transactions on Neural Networks*, 19(8):1431–1445, Aug 2008.
- [59] Cheng Zhang, Judith Bütepage, Hedvig Kjellström, and Stephan Mandt. Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):2008–2026, 2018.
- [60] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Balancing learning and inference in variational autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5885–5892, 2019.

Supplementary Material

A Computing the Latent Space Partition

In this section we first introduce notations and demonstrate how to express a region ω of the partition Ω as a polytope defined by a system of inequalities, and then leverage this formulation to demonstrate how to obtain Ω by recursively exploring neighboring regions starting from a random point/region.

Regions as Polytopes To represent the regions $\omega \in \Omega$ as a polytope via a system of inequalities we need to recall from (1) that the input-output mapping is defined on each region by the affine parameters A_ω, B_ω themselves obtained by composition of MASOs. Each layer pre-activation (feature map prior application of the nonlinearity) is denoted by $\mathbf{h}^\ell(\mathbf{z}) \in \mathbb{R}^{D^\ell}, \ell = 1, \dots, L-1$ and given by $\mathbf{h}^\ell(\mathbf{z}) = \mathbf{A}_\omega^{1 \rightarrow \ell} \mathbf{z} + \mathbf{b}_\omega^{1 \rightarrow \ell}$, with up-to-layer ℓ affine parameters

$$\mathbf{A}_\omega^{1 \rightarrow \ell} \triangleq \mathbf{W}^\ell \mathbf{D}_\omega^{\ell-1} \mathbf{W}^{\ell-1} \dots \mathbf{D}_\omega^1 \mathbf{W}^1, \quad \mathbf{A}_\omega^{1 \rightarrow \ell} \in \mathbb{R}^{D^\ell \times S}, \quad (10)$$

$$\mathbf{b}_\omega^{1 \rightarrow \ell-1} \triangleq \mathbf{v}^\ell + \sum_{i=1}^{\ell} \mathbf{W}^\ell \mathbf{D}_\omega^{\ell-1} \mathbf{W}^{\ell-1} \dots \mathbf{D}_\omega^i \mathbf{v}^i, \quad \mathbf{b}_\omega^{1 \rightarrow \ell} \in \mathbb{R}^{D^\ell}, \quad (11)$$

which depend on the region ω in the latent space². Notice that we have in particular $\mathbf{A}_\omega^L = \mathbf{A}_\omega$ and $\mathbf{b}_\omega^L = \mathbf{b}_\omega$, the entire DGN affine parameters from (2) on region ω . The regions depend on the signs of the pre-activations defined as $\mathbf{q}^\ell(\mathbf{z}) = \text{sign}(\mathbf{h}^\ell(\mathbf{z}))$ due to the used activation function behaving linearly as long as the feature maps preserve the same sign. This holds for (leaky-)ReLU or absolute value, for max-pooling we would need to look at the argmax position of each pooling window, as pooling is rare in DGN we focus here on DN without max-pooling; let $\mathbf{q}^{\text{all}}(\mathbf{z}) \triangleq [(\mathbf{q}^{L-1}(\mathbf{z}))^T, \dots, (\mathbf{q}^1(\mathbf{z}))^T]^T$ collect all the per layer sign operators without the last layer as it does not apply any activation.

Lemma 5. *The \mathbf{q}^{all} operator is piecewise constant and there is a bijection between Ω and $\text{Im}(\mathbf{q})$.*

The above demonstrates the equivalence of knowing ω in which an input \mathbf{z} belongs to and knowing the sign pattern of the feature maps associated to \mathbf{z} ; we will thus use interchangeably $\mathbf{q}^{\text{all}}(\mathbf{z}), \mathbf{z} \in \omega$ and $\mathbf{q}^{\text{all}}(\omega)$. From this, we see that the pre-activation signs and the regions are tied together. We can now leverage this result and provide the explicit region ω as a polytope via its system of inequality, to do so we need to collect the per-layer slopes and biases into

$$\mathbf{A}_\omega^{\text{all}} = \begin{bmatrix} \mathbf{A}_\omega^{1 \rightarrow L-1} \\ \vdots \\ \mathbf{A}_\omega^{1 \rightarrow 1} \end{bmatrix}, \quad \mathbf{b}_\omega^{\text{all}} = \begin{bmatrix} \mathbf{b}_\omega^{1 \rightarrow L-1} \\ \vdots \\ \mathbf{b}_\omega^{1 \rightarrow 1} \end{bmatrix}, \quad \mathbf{A}_\omega^{\text{all}} \in \mathbb{R}^{(\prod_{\ell=1}^{L-1} D^\ell) \times S}, \quad \mathbf{b}_\omega^{\text{all}} \in \mathbb{R}^{\prod_{\ell=1}^{L-1} D^\ell}. \quad (12)$$

Corollary 2. *The \mathcal{H} -representation of the polyhedral region ω is given by*

$$\omega = \{\mathbf{z} \in \mathbb{R}^S : \mathbf{A}_\omega^{\text{all}} \mathbf{z} < -\mathbf{q}^{\text{all}}(\omega) \odot \mathbf{b}_\omega^{\text{all}}\} = \bigcap_{\ell=1}^{L-1} \{\mathbf{z} \in \mathbb{R}^S : \mathbf{A}_\omega^{1 \rightarrow \ell} \mathbf{z} < -\mathbf{q}^\ell(\omega) \odot \mathbf{b}_\omega^{1 \rightarrow \ell}\}, \quad (13)$$

with \odot the Hadamard product.

From the above, it is clear that the sign locates in which side of each hyperplane the region is located. We now have a direct way to obtain the polytope ω from its sign pattern $\mathbf{q}^{\text{all}}(\omega)$ or equivalently from an input $\mathbf{z} \in \omega$; the only task left is to obtain the entire partition Ω collecting all the DN regions, which we now propose to do via a simple scheme.

Partition Cells Enumeration. The search for all cells in a partition is known as the cell enumeration problem and has been extensively studied in the context of specific partitions such as hyperplane arrangements [4, 50, 19]. In our case however, the set of inequalities of different regions changes. In fact, for any neighbour region, not only the sign pattern \mathbf{q}^{all} will change but also $\mathbf{A}_\omega^{\text{all}}$ and $\mathbf{b}_\omega^{\text{all}}$ due to

²looser condition can be put as the up-to-layer ℓ mapping is a CPA on a coarser partition than Ω but this is sufficient for our goal.

the composition of layers. In fact, changing one activation state say -1 to 1 for a specific unit at layer ℓ will alter the affine parameters from (10) and (11) due to the layer composition. As such, we propose to enumerate all the cells $\omega \in \Omega$ with a deterministic algorithm that starts from an initial region and recursively explores its neighbouring cells until all have been visited while recomputing the inequality system at each step. To do so, consider the initial region ω_0 . First, one finds all the non-redundant inequalities of the inequality system (13), the remaining inequalities define the faces of the polytope ω . Second, one obtains any of the neighbouring regions sharing a face with ω_0 by switching the sign in the entry of $\mathbf{q}(\omega_0)$ corresponding to the considered face. Repeat this for all non-redundant inequalities to obtain all the adjacent regions to ω_0 sharing a face with it. Each altered code defines an adjacent region and its system of inequality can be obtained as per Lemma 2. Doing so for all the faces of the initial region and then iterating this process on all the newly discovered regions will enumerate the entire partition Ω . We summarize this in Algo 1 in the appendix and illustrate this recursive procedure in Fig. 1.

We now have each cell as a polytope and enumerated the partition Ω , we can now turn into the computation of the marginal and posterior DGN distributions.

B Analytical Moments for truncated Gaussian

To lighten the derivation, we introduce extend the $[\cdot]$ indexing operator such that for example for a matrix, $[\cdot]_{-k, \cdot}$ means that all the rows but the k^{th} are taken, and all columns are taken. Also, $[\cdot]_{(k,l), \cdot}$ means that only the k^{th} and l^{th} rows are taken and all the columns. Let also introduce the following quantities

$$\begin{aligned} [F(\mathbf{a}, \Sigma)]_k &= \phi([\mathbf{a}]_k; 0, [\Sigma]_{k,k}) \Phi([\mathbf{a}]_{-k, \infty}) (\boldsymbol{\mu}(k), \Sigma(k)) \\ [G(\mathbf{a}, \Sigma)]_{k,l} &= \phi([\mathbf{a}]_{(k,l)}; 0, [\Sigma]_{(k,j),(k,j)}) \Phi([\mathbf{a}]_{-(k,l), \infty}) (\boldsymbol{\mu}((k,l)), \Sigma((k,l))) \\ H(\mathbf{a}, \Sigma) &= G(\mathbf{a}, \Sigma) + \text{diag} \left(\frac{\mathbf{l} \odot F(\mathbf{l}, \Sigma) - (\Sigma \odot G(\mathbf{l}, \Sigma)) \mathbf{1}}{\text{diag}(\Sigma)} \right) \end{aligned}$$

with $\boldsymbol{\mu}(u) = [\Sigma]_{-u,u} [\Sigma]_{u,u}^{-1} [\mathbf{a}]_u$, and $\Sigma(u) = [\Sigma]_{-u,-u} - [\Sigma]_{-u,u} [\Sigma]_{u,u}^{-1} [\Sigma]_{-u,u}^T$. Thanks to the above form, we can now obtain the integral $e_\omega^0(\Sigma) \triangleq \Phi_\omega(\mathbf{0}, \Sigma)$ and the first two moments of a centered truncated gaussian $e_\omega^1(\Sigma) \triangleq \int_\omega \mathbf{z} \phi(\mathbf{z}; \mathbf{0}, \Sigma)$ and $E_\omega^2(\Sigma) \triangleq \int_\omega \mathbf{z} \mathbf{z}^T \phi(\mathbf{z}; \mathbf{0}, \Sigma)$

Corollary 3. *The integral and first two moments of a centered truncated gaussian are given by*

$$e_\omega^0(\Sigma) = \sum_{\Delta \in T(\omega)} \sum_{(s,C) \in T(\Delta)} s \Phi_{[\mathbf{l}(C), \infty)}(0, R_C \Sigma R_C^T) dz, \quad (14)$$

$$e_\omega^1(\Sigma) = \Sigma \sum_{\Delta \in T(\omega)} \sum_{(s,C) \in T(\Delta)} s R_C^T F(\mathbf{l}_{\omega,C}, R_C \Sigma R_C^T), \quad (15)$$

$$E_\omega^2(\Sigma) = \Sigma \left(\sum_{\Delta \in T(\omega)} \sum_{(s,C) \in T(\Delta)} s R_C^T (H(\mathbf{l}_{\omega,C}, R_C \Sigma R_C^T)) R_C \right) \Sigma + e_\omega^0(\Sigma) \Sigma \quad (16)$$

To simplify notations let consider the following notation of the posterior (6) where we incorporate the terms independent of \mathbf{z} into

$$\alpha_\omega(\mathbf{x}) = \frac{\phi(\mathbf{x}; B_\omega, \Sigma_\mathbf{x} + A_\omega \Sigma_\mathbf{z} A_\omega^T)}{\sum_\omega \phi(\mathbf{x}; B_\omega, \Sigma_\mathbf{x} + A_\omega \Sigma_\mathbf{z} A_\omega^T) \Phi_\omega(\boldsymbol{\mu}_\omega(\mathbf{x}), \Sigma_\omega)}, \quad (17)$$

leading to $p(\mathbf{z}|\mathbf{x}) = \sum_{\omega \in \Omega} \delta_\omega(\mathbf{z}) \alpha_\omega(\mathbf{x}) \phi(\mathbf{z}; \boldsymbol{\mu}_\omega(\mathbf{x}), \Sigma_\omega)$.

Theorem 2. *The first (per region) moments of the DGN posterior are given by*

$$\begin{aligned} \mathbb{E}_{\mathbf{z}|\mathbf{x}}[\mathbb{1}_{\mathbf{z} \in \omega}] &= \alpha_\omega(\mathbf{x}) e_\omega^0(\Sigma_\omega), \\ \mathbb{E}_{\mathbf{z}|\mathbf{x}}[\mathbf{z} \mathbb{1}_{\mathbf{z} \in \omega}] &= \alpha_\omega(\mathbf{x}) (e_{\omega - \boldsymbol{\mu}_\omega(\mathbf{x})}^1(\Sigma_\omega) + e_{\omega - \boldsymbol{\mu}_\omega(\mathbf{x})}^0(\Sigma_\omega) \boldsymbol{\mu}_\omega(\mathbf{x})) \end{aligned}$$

$$\begin{aligned}\mathbb{E}_{\mathbf{z}|\mathbf{x}}[\mathbf{z}\mathbf{z}^T \mathbb{1}_{\mathbf{z}\in\omega}] &= \alpha_\omega(\mathbf{x})(\mathbf{E}_{\omega-\boldsymbol{\mu}_\omega(\mathbf{x})}^2(\boldsymbol{\Sigma}_\omega) + \mathbf{e}_{\omega-\boldsymbol{\mu}_\omega(\mathbf{x})}^1(\boldsymbol{\Sigma}_\omega)\boldsymbol{\mu}_\omega(\mathbf{x})^T \\ &\quad + \boldsymbol{\mu}_{\omega-\boldsymbol{\mu}_\omega(\mathbf{x})}(\mathbf{x})\mathbf{e}_{\omega-\boldsymbol{\mu}_\omega(\mathbf{x})}^1(\mathbf{x})^T + \boldsymbol{\mu}_\omega(\mathbf{x})\boldsymbol{\mu}_\omega(\mathbf{x})^T \mathbf{e}_\omega^0(\mathbf{x}))\end{aligned}$$

which we denote $\mathbb{E}_{\mathbf{z}|\mathbf{x}}[\mathbb{1}_{\mathbf{z}\in\omega}] \triangleq \mathbf{e}_\omega^0(\mathbf{x})$, $\mathbb{E}_{\mathbf{z}|\mathbf{x}}[\mathbf{z} \mathbb{1}_{\mathbf{z}\in\omega}] \triangleq \mathbf{e}_\omega^1(\mathbf{x})$ and $\mathbb{E}_{\mathbf{z}|\mathbf{x}}[\mathbf{z}\mathbf{z}^T \mathbb{1}_{\mathbf{z}\in\omega}] \triangleq \mathbf{E}_\omega^2(\mathbf{x})$. (Proof in F.9.)

C Implementation Details

The Delaunay triangulation needs the \mathcal{V} -representation of ω , the vertices which convex hull form the region [24]. Given that we have the \mathcal{H} -representation, finding the vertices is known as the vertex enumeration problem [18]. To compute the triangulation we use the Python scipy [57] implementation which interfaces the C/C++ Qhull implementation [8]. To compute the $\mathcal{H} \mapsto \mathcal{V}$ representation and vice-versa we leverage pycddlib³ which interfaces the C/C++ cddlib library⁴ employing the double description method [43].

D Figures

We demonstrate here additional figures for the posterior and marginal distribution of a DGN.

³<https://pypi.org/project/pycddlib/>

⁴https://inf.ethz.ch/personal/fukudak/cdd_home/index.html

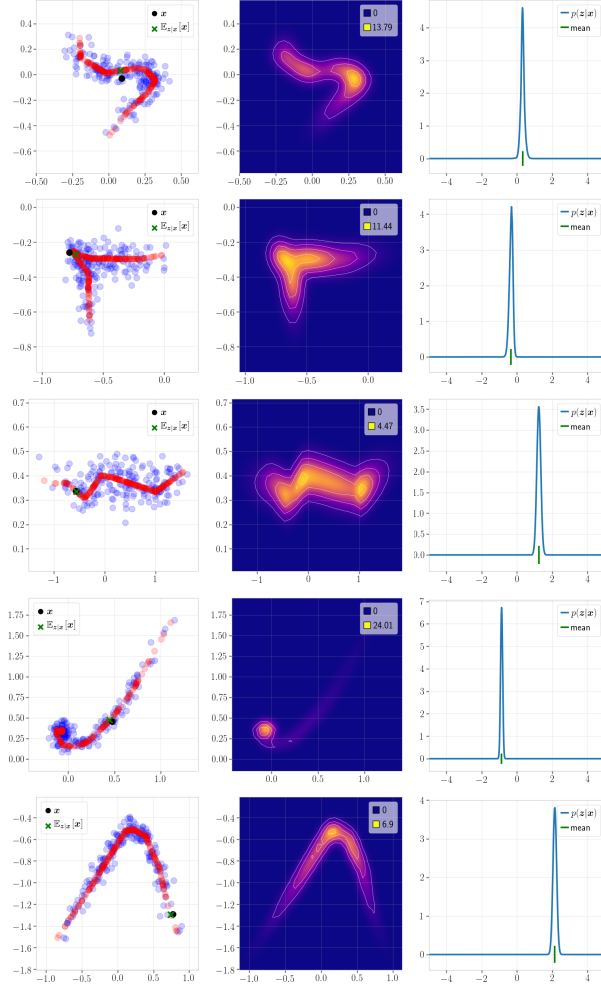


Figure 6: Additional random DGNs with their samples, the posterior and the marginal distributions.

E Algorithms

Algorithm 1: SearchRegion

Data: Starting region ω and $q(\omega)$, initial set (Ω)

Result: Updated Ω

if $\omega \notin \Omega$ **then**

$\Omega \leftarrow \Omega \cup \{\omega\};$

else

 Quit

end

$I = \text{reduce}(A_\omega^{\text{all}}, B_\omega^{\text{all}});$

for $i \in I$ **do**

 SearchRegion(flip($q(\omega)$), i), Ω);

end

F Proofs

In this section we provide all the proofs for the main paper theoretical claims. In particular we will go through the derivations of the per region posterior first moments and then the derivation of the expectation and maximization steps.

F.1 Proof of Lemma 1

Proof. The proof consists of expressing the conditional distribution and using the properties of DGN with piecewise affine nonlinearities. We are able to split the distribution into a mixture model as follows:

$$\begin{aligned}
 p(\mathbf{x}|\mathbf{z}) &= \frac{1}{(2\pi)^{D/2} \sqrt{|\det \Sigma_{\mathbf{x}}|}} e^{-\frac{1}{2}(\mathbf{x}-g(\mathbf{z}))^T \Sigma_{\mathbf{x}}^{-1}(\mathbf{x}-g(\mathbf{z}))} \\
 &= \frac{1}{(2\pi)^{D/2} \sqrt{|\det \Sigma_{\mathbf{x}}|}} e^{-\frac{1}{2}(\mathbf{x}-\sum_{\omega \in \Omega} \mathbb{1}_{\mathbf{z} \in \omega} (A_\omega \mathbf{z} + B_\omega))^T \Sigma_{\mathbf{x}}^{-1}(\mathbf{x}-\sum_{\omega \in \Omega} \mathbb{1}_{\mathbf{z} \in \omega} (A_\omega \mathbf{z} + B_\omega))} \\
 &= \frac{1}{(2\pi)^{D/2} \sqrt{|\det \Sigma_{\mathbf{x}}|}} e^{-\frac{1}{2} \sum_{\omega \in \Omega} \mathbb{1}_{\mathbf{z} \in \omega} (\mathbf{x} - (A_\omega \mathbf{z} + B_\omega))^T \Sigma_{\mathbf{x}}^{-1}(\mathbf{x} - (A_\omega \mathbf{z} + B_\omega))} \\
 &= \sum_{\omega \in \Omega} \mathbb{1}_{\mathbf{z} \in \omega} \frac{1}{(2\pi)^{D/2} \sqrt{|\det \Sigma_{\mathbf{x}}|}} e^{-\frac{1}{2}(\mathbf{x} - (A_\omega \mathbf{z} + B_\omega))^T \Sigma_{\mathbf{x}}^{-1}(\mathbf{x} - (A_\omega \mathbf{z} + B_\omega))} \\
 &= \sum_{\omega \in \Omega} \mathbb{1}_{\mathbf{z} \in \omega} \phi(\mathbf{x} | A_\omega \mathbf{z} + B_\omega, \Sigma_{\mathbf{x}})
 \end{aligned}$$

□

F.2 Proof of Proposition 1

Proof. This result is direct by noticing that the probability to obtain a specific region slope and bias is the probability that the sampled latent vector lies in the corresponding region. This probability is obtained simply by integrating the latent gaussian distribution on the region. We obtain the result of the proposition. □

F.3 Proof of Theorem 1

Proof. For the first part, we simply leverage the known result from linear Gaussian models [48] stating that

$$\begin{aligned}
p(\mathbf{z}|\mathbf{x}) &= \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})} \\
&= \frac{1}{p(\mathbf{x})} \frac{e^{-\frac{1}{2}(\mathbf{x}-g(\mathbf{z}))^T \Sigma_{\mathbf{x}}^{-1}(\mathbf{x}-g(\mathbf{z}))}}{(2\pi)^{D/2} \sqrt{|\det(\Sigma_{\mathbf{x}})|}} \frac{e^{-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^T \Sigma_{\mathbf{z}}^{-1}(\mathbf{z}-\boldsymbol{\mu})}}{(2\pi)^{S/2} \sqrt{|\det(\Sigma_{\mathbf{z}})|}} \\
&= \frac{1}{p(\mathbf{x})} \left(\sum_{\omega \in \Omega} \mathbb{1}_{\mathbf{z} \in \omega} \frac{e^{-\frac{1}{2}(\mathbf{x}-A_{\omega}\mathbf{z}-B_{\omega})^T \Sigma_{\mathbf{x}}^{-1}(\mathbf{x}-A_{\omega}\mathbf{z}-B_{\omega})}}{(2\pi)^{D/2} \sqrt{|\det(\Sigma_{\mathbf{x}})|}} \right) \frac{e^{-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^T \Sigma_{\mathbf{z}}^{-1}(\mathbf{z}-\boldsymbol{\mu})}}{(2\pi)^{S/2} \sqrt{|\det(\Sigma_{\mathbf{z}})|}} \\
&= \frac{1}{p(\mathbf{x})} \sum_{\omega \in \Omega} \mathbb{1}_{\mathbf{z} \in \omega} \frac{e^{-\frac{1}{2}(\mathbf{x}-A_{\omega}\mathbf{z}-B_{\omega})^T \Sigma_{\mathbf{x}}^{-1}(\mathbf{x}-A_{\omega}\mathbf{z}-B_{\omega}) - \frac{1}{2}\mathbf{z}^T \Sigma_{\mathbf{z}}^{-1} \mathbf{z}}}{(2\pi)^{(S+D)/2} \sqrt{|\det(\Sigma_{\mathbf{x}})| |\det(\Sigma_{\mathbf{z}})|}} \\
&= \frac{1}{p(\mathbf{x})} \sum_{\omega \in \Omega} \mathbb{1}_{\mathbf{z} \in \omega} \frac{e^{-\frac{1}{2}((\mathbf{x}-B_{\omega})-A_{\omega}\mathbf{z})^T \Sigma_{\mathbf{x}}^{-1}((\mathbf{x}-B_{\omega})-A_{\omega}\mathbf{z}) - \frac{1}{2}\mathbf{z}^T \Sigma_{\mathbf{z}}^{-1} \mathbf{z}}}{(2\pi)^{(S+D)/2} \sqrt{|\det(\Sigma_{\mathbf{x}})| |\det(\Sigma_{\mathbf{z}})|}} \\
&= \frac{1}{p(\mathbf{x})} \sum_{\omega \in \Omega} \mathbb{1}_{\mathbf{z} \in \omega} \frac{e^{-\frac{1}{2}((A_{\omega}^T \Sigma_{\mathbf{x}}^{-1} A_{\omega} + \Sigma_{\mathbf{z}}^{-1})^{-1} A_{\omega}^T \Sigma_{\mathbf{x}}^{-1}(\mathbf{x}-B_{\omega}) - \mathbf{z})^T (A_{\omega}^T \Sigma_{\mathbf{x}}^{-1} A_{\omega} + \Sigma_{\mathbf{z}}^{-1})((A_{\omega}^T \Sigma_{\mathbf{x}}^{-1} A_{\omega} + \Sigma_{\mathbf{z}}^{-1})^{-1} A_{\omega}^T \Sigma_{\mathbf{x}}^{-1}(\mathbf{x}-B_{\omega}) - \mathbf{z})}}{(2\pi)^{(S+D)/2} \sqrt{|\det(\Sigma_{\mathbf{x}})| |\det(\Sigma_{\mathbf{z}})|}} \\
&\quad \times e^{-\frac{1}{2}((\mathbf{x}-B_{\omega})^T \Sigma_{\mathbf{x}}^{-1}(\mathbf{x}-B_{\omega})) + \frac{1}{2}((\mathbf{x}-B_{\omega})^T \Sigma_{\mathbf{x}}^{-1} A_{\omega} (A_{\omega}^T \Sigma_{\mathbf{x}}^{-1} A_{\omega} + \Sigma_{\mathbf{z}}^{-1})^{-1} A_{\omega}^T \Sigma_{\mathbf{x}}^{-1}(\mathbf{x}-B_{\omega}))} \\
&= \frac{1}{p(\mathbf{x})} \sum_{\omega \in \Omega} \mathbb{1}_{\mathbf{z} \in \omega} \frac{e^{-\frac{1}{2}((A_{\omega}^T \Sigma_{\mathbf{x}}^{-1} A_{\omega} + \Sigma_{\mathbf{z}}^{-1})^{-1} A_{\omega}^T \Sigma_{\mathbf{x}}^{-1}(\mathbf{x}-B_{\omega}) - \mathbf{z})^T (A_{\omega}^T \Sigma_{\mathbf{x}}^{-1} A_{\omega} + \Sigma_{\mathbf{z}}^{-1})((A_{\omega}^T \Sigma_{\mathbf{x}}^{-1} A_{\omega} + \Sigma_{\mathbf{z}}^{-1})^{-1} A_{\omega}^T \Sigma_{\mathbf{x}}^{-1}(\mathbf{x}-B_{\omega}) - \mathbf{z})}}{(2\pi)^{(S+D)/2} \sqrt{|\det(\Sigma_{\mathbf{x}})| |\det(\Sigma_{\mathbf{z}})|}} \\
&\quad \times e^{-\frac{1}{2}((\mathbf{x}-B_{\omega})^T (\Sigma_{\mathbf{x}}^{-1} - \Sigma_{\mathbf{x}}^{-1} A_{\omega} (A_{\omega}^T \Sigma_{\mathbf{x}}^{-1} A_{\omega} + \Sigma_{\mathbf{z}}^{-1})^{-1} A_{\omega}^T \Sigma_{\mathbf{x}}^{-1})(\mathbf{x}-B_{\omega}))} \\
&= \frac{1}{p(\mathbf{x})} \sum_{\omega \in \Omega} \mathbb{1}_{\mathbf{z} \in \omega} \frac{e^{-\frac{1}{2}((A_{\omega}^T \Sigma_{\mathbf{x}}^{-1} A_{\omega} + \Sigma_{\mathbf{z}}^{-1})^{-1} A_{\omega}^T \Sigma_{\mathbf{x}}^{-1}(\mathbf{x}-B_{\omega}) - \mathbf{z})^T (A_{\omega}^T \Sigma_{\mathbf{x}}^{-1} A_{\omega} + \Sigma_{\mathbf{z}}^{-1})((A_{\omega}^T \Sigma_{\mathbf{x}}^{-1} A_{\omega} + \Sigma_{\mathbf{z}}^{-1})^{-1} A_{\omega}^T \Sigma_{\mathbf{x}}^{-1}(\mathbf{x}-B_{\omega}) - \mathbf{z})}}{(2\pi)^{(S+D)/2} \sqrt{|\det(\Sigma_{\mathbf{x}})| |\det(\Sigma_{\mathbf{z}})|}} \\
&\quad \times e^{-\frac{1}{2}((\mathbf{x}-B_{\omega})^T (\Sigma_{\mathbf{x}} + A_{\omega} \Sigma_{\mathbf{z}} A_{\omega}^T)^{-1}(\mathbf{x}-B_{\omega}))} \\
&= \frac{1}{p(\mathbf{x})} \sum_{\omega \in \Omega} \mathbb{1}_{\mathbf{z} \in \omega} \frac{e^{-\frac{1}{2}(\boldsymbol{\mu}_{\omega}(\mathbf{x}) - \mathbf{z})^T \Sigma_{\omega}^{-1}(\boldsymbol{\mu}_{\omega}(\mathbf{x}) - \mathbf{z})}}{(2\pi)^{(S+D)/2} \sqrt{|\det(\Sigma_{\mathbf{x}})| |\det(\Sigma_{\mathbf{z}})|}} e^{-\frac{1}{2}((\mathbf{x}-B_{\omega})^T (\Sigma_{\mathbf{x}} + A_{\omega} \Sigma_{\mathbf{z}} A_{\omega}^T)^{-1}(\mathbf{x}-B_{\omega}))}
\end{aligned}$$

with $\boldsymbol{\mu}_{\omega}(\mathbf{x}) = \Sigma_{\omega} A_{\omega}^T \Sigma_{\mathbf{x}}^{-1}(\mathbf{x} - B_{\omega})$ and $\Sigma_{\omega} = (A_{\omega}^T \Sigma_{\mathbf{x}}^{-1} A_{\omega} + \Sigma_{\mathbf{z}}^{-1})^{-1}$ as a result it corresponds to a mixture of truncated gaussian, each living on ω . Now we determine the renormalization constant:

$$\begin{aligned}
p(\mathbf{x}) &= \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \\
&= \sum_{\omega \in \Omega} \int_{\omega} \mathbb{1}_{\mathbf{z} \in \omega} \frac{e^{-\frac{1}{2}(\boldsymbol{\mu}_{\omega}(\mathbf{x}) - \mathbf{z})^T \Sigma_{\omega}^{-1}(\boldsymbol{\mu}_{\omega}(\mathbf{x}) - \mathbf{z})}}{(2\pi)^{(S+D)/2} \sqrt{|\det(\Sigma_{\mathbf{x}})| |\det(\Sigma_{\mathbf{z}})|}} e^{-\frac{1}{2}((\mathbf{x}-B_{\omega})^T (\Sigma_{\mathbf{x}} + A_{\omega} \Sigma_{\mathbf{z}} A_{\omega}^T)^{-1}(\mathbf{x}-B_{\omega}))} d\mathbf{z} \\
&= \sum_{\omega \in \Omega} \mathbb{1}_{\mathbf{z} \in \omega} \frac{e^{-\frac{1}{2}((\mathbf{x}-B_{\omega})^T (\Sigma_{\mathbf{x}} + A_{\omega} \Sigma_{\mathbf{z}} A_{\omega}^T)^{-1}(\mathbf{x}-B_{\omega}))}}{(2\pi)^{D/2} \sqrt{|\det(\Sigma_{\mathbf{x}})| |\det(\Sigma_{\mathbf{z}})|}} \sqrt{\det(\Sigma_{\omega})} \int_{\omega} \phi(\mathbf{z}; \boldsymbol{\mu}_{\omega}(\mathbf{x}), \Sigma_{\omega}) d\mathbf{z} \\
&= \sum_{\omega \in \Omega} \mathbb{1}_{\mathbf{z} \in \omega} \frac{e^{-\frac{1}{2}((\mathbf{x}-B_{\omega})^T (\Sigma_{\mathbf{x}} + A_{\omega} \Sigma_{\mathbf{z}} A_{\omega}^T)^{-1}(\mathbf{x}-B_{\omega}))}}{(2\pi)^{D/2} \sqrt{|\det(\Sigma_{\mathbf{x}})| |\det(\Sigma_{\mathbf{z}})|}} \sqrt{\det(\Sigma_{\omega})} \Phi_{\omega}(\boldsymbol{\mu}_{\omega}(\mathbf{x}), \Sigma_{\omega})
\end{aligned}$$

$$= \sum_{\omega \in \Omega} \mathbb{1}_{\mathbf{z} \in \omega} \frac{\sqrt{\det(\Sigma_{\mathbf{x}} + A_{\omega} \Sigma_{\mathbf{z}} A_{\omega}^T) \det(\Sigma_{\omega})}}{\sqrt{|\det(\Sigma_{\mathbf{x}})| |\det(\Sigma_{\mathbf{z}})|}} \phi(\mathbf{x}; B_{\omega}, \Sigma_{\mathbf{x}} + A_{\omega} \Sigma_{\mathbf{z}} A_{\omega}^T) \Phi_{\omega}(\boldsymbol{\mu}_{\omega}(\mathbf{x}), \Sigma_{\omega}),$$

now using the Matrix determinant lemma [26] we have that $\det(\Sigma_{\mathbf{x}} + A_{\omega} \Sigma_{\mathbf{z}} A_{\omega}^T) = \det(\Sigma_{\mathbf{z}}^{-1} + A_{\omega}^T \Sigma_{\mathbf{x}}^{-1} A_{\omega}) \det(\Sigma_{\mathbf{x}}) \det(\Sigma_{\mathbf{z}})$ leading to

$$p(\mathbf{x}) = \sum_{\omega} \phi(\mathbf{x}; B_{\omega}, \Sigma_{\mathbf{x}} + A_{\omega} \Sigma_{\mathbf{z}} A_{\omega}^T) \Phi_{\omega}(\boldsymbol{\mu}_{\omega}(\mathbf{x}), \Sigma_{\omega}),$$

$$p(\mathbf{z}|\mathbf{x}) = \sum_{\omega} \delta_{\omega}(\mathbf{z}) \frac{\phi(\mathbf{x}; B_{\omega}, \Sigma_{\mathbf{x}} + A_{\omega} \Sigma_{\mathbf{z}} A_{\omega}^T) \phi(\mathbf{z}; \boldsymbol{\mu}_{\omega}(\mathbf{x}), \Sigma_{\omega})}{\sum_{\omega} \phi(\mathbf{x}; B_{\omega}, \Sigma_{\mathbf{x}} + A_{\omega} \Sigma_{\mathbf{z}} A_{\omega}^T) \Phi_{\omega}(\boldsymbol{\mu}_{\omega}(\mathbf{x}), \Sigma_{\omega})}.$$

□

F.4 Proof of Lemma 2

The proof will consist of observing that the posterior (prior rewriting) can be expressed as a softmax of a quantity rescaled by the standard deviation.

Proof.

$$\begin{aligned} \log(\phi(\mathbf{z}; \boldsymbol{\mu}_{\omega}(\mathbf{x}), \Sigma_{\omega})) &= -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_{\omega})^T \Sigma_{\omega}^{-1}(\mathbf{x} - \boldsymbol{\mu}_{\omega}) - \frac{1}{2} \log(\det(\Sigma_{\omega})) + cst \\ &= -\frac{1}{2}(\mathbf{z} - \Sigma_{\omega} A_{\omega}^T \Sigma_{\mathbf{x}}^{-1}(A_0 \mathbf{z}_0 + B_0 - B_{\omega}))^T \Sigma_{\omega}^{-1}(\mathbf{z} - \Sigma_{\omega} A_{\omega}^T \Sigma_{\mathbf{x}}^{-1}(A_0 \mathbf{z}_0 + B_0 - B_{\omega}))^T \\ &\quad - \frac{1}{2} \log(\det(\Sigma_{\omega})) + cst \\ &= -\frac{1}{2}(\mathbf{z} - (A_{\omega}^T \Sigma_{\mathbf{x}}^{-1} A_{\omega})^{-1} A_{\omega}^T \Sigma_{\mathbf{x}}^{-1}(A_0 \mathbf{z}_0 + B_0 - B_{\omega}))^T \Sigma_{\omega}^{-1}(\mathbf{z} - (A_{\omega}^T \Sigma_{\mathbf{x}}^{-1} A_{\omega})^{-1} A_{\omega}^T \Sigma_{\mathbf{x}}^{-1}(A_0 \mathbf{z}_0 + B_0 - B_{\omega}))^T \\ &\quad - \frac{1}{2} \log(\det(\Sigma_{\omega})) + cst \end{aligned}$$

where we used the following result to develop $\boldsymbol{\mu}_{\omega}(\mathbf{x})$

$$\begin{aligned} \Sigma_{\omega} &= (A_{\omega}^T \Sigma_{\mathbf{x}}^{-1} A_{\omega} + \Sigma_{\mathbf{z}}^{-1})^{-1} = (A_{\omega}^T \Sigma_{\mathbf{x}}^{-1} A_{\omega} + (A_{\omega}^T \Sigma_{\mathbf{x}}^{-1} A_{\omega})(A_{\omega}^T \Sigma_{\mathbf{x}}^{-1} A_{\omega})^{-1} \Sigma_{\mathbf{z}}^{-1})^{-1} \\ &= (A_{\omega}^T \Sigma_{\mathbf{x}}^{-1} A_{\omega})^{-1} (I + (A_{\omega}^T \Sigma_{\mathbf{x}}^{-1} A_{\omega})^{-1} \Sigma_{\mathbf{z}}^{-1})^{-1} \\ &= (A_{\omega}^T \Sigma_{\mathbf{x}}^{-1} A_{\omega})^{-1} \text{ as } (\Sigma_{\mathbf{z}} A_{\omega}^T \Sigma_{\mathbf{x}}^{-1} A_{\omega})^{-1} \rightarrow \mathbf{0}. \end{aligned}$$

if we are in the same region ω than \mathbf{z}_0 then the above becomes

$$\arg \max_{\mathbf{z} \in \omega_0} \log(\phi(\mathbf{z}; \boldsymbol{\mu}_{\omega}(\mathbf{x}), \Sigma_{\omega})) = \arg \max_{\mathbf{z} \in \omega_0} -\frac{1}{2}(\mathbf{z} - \mathbf{z}_0)^T \Sigma_{\omega}^{-1}(\mathbf{z} - \mathbf{z}_0) = \mathbf{z}_0,$$

and since we know that we are in the same region, the argmax $\mathbf{z} = \mathbf{z}_0$ lies in this region and thus is the maximum of the posterior.

□

F.5 Proof of Lemma 3

Proof. The sign vectors represent the sign of each pre-activation feature maps. The key here is that when changing the sign, the input passes through the knot of the corresponding activation function of that layer. This implies a change in the region in the DGN input space. In fact, without degenerate weights and with nonzero activation functions, a change in any dimension of the sign vector (used to

form the per region slope and bias) impact a change in the affine mapping used to map inputs z to outputs x . As such, whenever a sign changes, the affine mapping changes, leading to a change of region in the DGN input space. As the sign vector is formed from the DGN input space, and we restrict ourselves to the image of this mapping, there does not exist a sign pattern/configuration not reachable by the DGN (otherwise it would not be in the image of this mapping). Now for the other inclusion, recall that a change in region and thus in per region affine mapping can only occur with a change of pre-activation sign pattern. \square

F.6 Proof of Corollary 2

Proof. From the above result, it is clear that the preactivation roots define the boundaries of the regions. Obtaining the hyperplane representation of the region thus simply consists of reexpressing this statement with the explicit pre-activation hyperplanes for all the layers and units, the intersection between layers coming from the subdivision. For additional details please see [6]. \square

F.7 Proof of Lemma 4

Proof. The proof consists of rearranging the terms from the inclusion-exclusion formula as in

$$\begin{aligned}
& \sum_{J \subseteq \{1, \dots, F\}, J \neq \emptyset} (-1)^{|J|+1} (\cap_{j \in J} A_j) = \cup_i A_i \\
(-1)^{F+1} S + & \sum_{J \subseteq \{1, \dots, F\}, J \neq \emptyset, |J| < F} (-1)^{|J|+1} (\cap_{j \in J} A_j) = \cup_i A_i \\
& (-1)^{F+1} S = \cup_i A_i - \sum_{J \subseteq \{1, \dots, F\}, J \neq \emptyset, |J| < F} (-1)^{|J|+1} (\cap_{j \in J} A_j) \\
& S = (-1)^{F+1} \cup_i A_i - (-1)^{F+1} \sum_{J \subseteq \{1, \dots, F\}, J \neq \emptyset, |J| < F} (-1)^{|J|+1} (\cap_{j \in J} A_j) \\
& S = (-1)^{F+1} \cup_i A_i + \sum_{J \subseteq \{1, \dots, F\}, J \neq \emptyset, |J| < F} (-1)^{|J|+1+F} (\cap_{j \in J} A_j)
\end{aligned}$$

then by application of Chasles rule [28], the integral domain can be decomposed into the signed sum of per cone integration. Finally, a simplex in dimension S has $S + 1$ faces, making $F = S + 1$ and leading to the desired result. \square

F.8 Proof of Moments

Lemma 6. *The first moments of Gaussian integration on an open rectangle defined by its lower limits \mathbf{a} is given by*

$$\int_{\mathbf{a}}^{\infty} \mathbf{z} \phi(\mathbf{0}, \Sigma) d\mathbf{z} = \Sigma F(\mathbf{a}), \quad (18)$$

$$\int_{\mathbf{a}}^{\infty} \mathbf{z} \mathbf{z}^T \phi(\mathbf{0}, \Sigma) d\mathbf{z} = \Phi_{[\mathbf{a}, \infty)}(\mathbf{0}, \Sigma) \Sigma + \Sigma \left(G(\mathbf{a}) + \frac{\mathbf{a} \odot F(\mathbf{a}) - (\Sigma \odot G(\mathbf{a})) \mathbf{1}}{\text{diag}(\Sigma)} \right) \Sigma. \quad (19)$$

where the division is performed elementwise.

Proof. First moment:

$$\begin{aligned}
\int_{\omega} \mathbf{z} \phi(\mathbf{x}; \mathbf{0}, \Sigma) d\mathbf{z} &= \int_{\omega} \mathbf{z} \frac{e^{-\frac{1}{2} \mathbf{z}^T \Sigma^{-1} \mathbf{z}}}{(2\pi)^{K/2} |\det(\Sigma)|^{1/2}} d\mathbf{z} \\
&= \sum_{\Delta \in S(\omega)} \sum_{(s, C) \in T(\Delta)} s \int_C \mathbf{z} \frac{e^{-\frac{1}{2} (R_C \mathbf{z})^T (R_C^T)^{-1} \Sigma_{\omega}^{-1} R_C^{-1} R_C \mathbf{z}}}{(2\pi)^{K/2} |\det(\Sigma_{\omega})|^{1/2}} d\mathbf{z}
\end{aligned}$$

$$\begin{aligned}
&= \sum_{\Delta \in S(\omega)} \sum_{(s,C) \in T(\Delta)} s \int_{\mathbf{l}(C)} R^{-1} \mathbf{u} \frac{e^{-\frac{1}{2} \mathbf{u}^T (R_C \Sigma_\omega R_C^T)^{-1} \mathbf{u}}}{(2\pi)^{K/2} |\det(R_C)| |\det(\Sigma_\omega)|^{1/2}} d\mathbf{u} \\
&= \sum_{\Delta \in S(\omega)} \sum_{(s,C) \in T(\Delta)} s R_C^{-1} \int_{\mathbf{l}(C)} \mathbf{u} \phi(\mathbf{u}; \mathbf{0}, R_C \Sigma_\omega R_C^T) d\mathbf{u} \\
&= \sum_{\Delta \in S(\omega)} \sum_{(s,C) \in T(\Delta)} s R_C^{-1} (R_C \Sigma_\omega R_C^T F(\mathbf{l}(C))) \\
&= \Sigma_\omega \sum_{\Delta \in S(\omega)} \sum_{(s,C) \in T(\Delta)} s R_C^T F(\mathbf{l}(C))
\end{aligned}$$

Second moment

$$\begin{aligned}
\int_{\omega} \mathbf{z} \mathbf{z}^T \phi(\mathbf{x}; \mathbf{0}, \Sigma) d\mathbf{z} &= \int_{\omega} \mathbf{z} \mathbf{z}^T \frac{e^{-\frac{1}{2} \mathbf{z}^T \Sigma^{-1} \mathbf{z}}}{(2\pi)^{K/2} |\det(\Sigma)|^{1/2}} d\mathbf{z} \\
&= \sum_{\Delta \in S(\omega)} \sum_{(s,C) \in T(\Delta)} s \int_C \mathbf{z} \mathbf{z}^T \frac{e^{-\frac{1}{2} (R_C \mathbf{y})^T (R_C^T)^{-1} \Sigma_\omega^{-1} R_C^{-1} R_C \mathbf{y}}}{(2\pi)^{K/2} |\det(\Sigma_\omega)|^{1/2}} d\mathbf{z} \\
&= \sum_{\Delta \in S(\omega)} \sum_{(s,C) \in T(\Delta)} s \int_{\mathbf{l}(C)} R_C^{-1} \mathbf{u} \mathbf{u}^T (R_C^{-1})^T \frac{e^{-\frac{1}{2} \mathbf{u}^T (R_C \Sigma_\omega R_C^T)^{-1} \mathbf{u}}}{(2\pi)^{K/2} |\det(R_C)| |\det(\Sigma_\omega)|^{1/2}} d\mathbf{u} \\
&= \sum_{\Delta \in S(\omega)} \sum_{(s,C) \in T(\Delta)} s R_C^{-1} \int_{\mathbf{l}(C)} \mathbf{u} \mathbf{u}^T \phi(\mathbf{u}; \mathbf{0}, R_C \Sigma_\omega R_C^T) d\mathbf{u} (R_C^{-1})^T \\
&= \sum_{\Delta \in S(\omega - \boldsymbol{\mu}_\omega(\mathbf{x}))} \sum_{(s,C) \in T(\Delta)} s R_C^{-1} \left[\Phi_{[\mathbf{l}(C), \infty)}(\mathbf{0}, R_C \Sigma_\omega R_C^T) R_C \Sigma_\omega R_C^T \right. \\
&\quad \left. + R_C \Sigma_\omega R_C^T \left(\frac{\mathbf{l}(C) \odot F(\mathbf{l}(C)) + (R_C \Sigma_\omega R_C^T \odot G(\mathbf{l}(C))) \mathbf{1}}{\text{diag}(R_C \Sigma_\omega R_C^T)} \right) (R_C \Sigma_\omega R_C^T)^T \right] (R_C^{-1})^T \\
&= \sum_{\Delta \in S(\omega - \boldsymbol{\mu}_\omega(\mathbf{x}))} \sum_{(s,C) \in T(\Delta)} s \left[\Phi_{[\mathbf{l}(C), \infty)}(\mathbf{0}, R_C \Sigma_\omega R_C^T) \Sigma_\omega \right. \\
&\quad \left. + \Sigma_\omega R_C^T \left(\frac{\mathbf{l}(C) \odot F(\mathbf{l}(C)) + (R_C \Sigma_\omega R_C^T \odot G(\mathbf{l}(C))) \mathbf{1}}{\text{diag}(R_C \Sigma_\omega R_C^T)} \right) R_C \Sigma_\omega \right] \\
&= e_{\omega - \boldsymbol{\mu}_\omega(\mathbf{x})}^0 \Sigma_\omega + \Sigma_\omega \left[\sum_{\Delta \in S(\omega - \boldsymbol{\mu}_\omega(\mathbf{x}))} \sum_{(s,C) \in T(\Delta)} s R_C^T \left(\frac{\mathbf{l}(C) \odot F(\mathbf{l}(C)) + (R_C \Sigma_\omega R_C^T \odot G(\mathbf{l}(C))) \mathbf{1}}{\text{diag}(R_C \Sigma_\omega R_C^T)} \right) R_C \right] \Sigma_\omega
\end{aligned}$$

□

F.9 Proof of Theorem 2

Proof. Constant: $R_C = \begin{pmatrix} C^T \\ H^T \Sigma_\omega^{-1} \end{pmatrix}$

$$\int_{\omega} p(\mathbf{z}|\mathbf{x}) d\mathbf{z} = \alpha_\omega(\mathbf{x}) \int_{\omega} \phi(\mathbf{z}; \boldsymbol{\mu}_\omega(\mathbf{x}), \Sigma_\omega) d\mathbf{z} = \alpha_\omega(\mathbf{x}) \int_{\omega - \boldsymbol{\mu}_\omega(\mathbf{x})} \phi(\mathbf{z}; \mathbf{0}, \Sigma_\omega) d\mathbf{z} = \alpha_\omega(\mathbf{x}) e_{\omega - \boldsymbol{\mu}_\omega(\mathbf{x})}^0$$

First moment:

$$\begin{aligned}
\int_{\omega} \mathbf{z} p(\mathbf{z}|\mathbf{x}) d\mathbf{z} &= \alpha_{\omega}(\mathbf{x}) \int_{\omega} \mathbf{z} \frac{e^{-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu}_{\omega}(\mathbf{x}))^T \Sigma_{\omega}^{-1}(\mathbf{z}-\boldsymbol{\mu}_{\omega}(\mathbf{x}))}}{(2\pi)^{K/2} |\det(\Sigma_{\omega})|^{1/2}} d\mathbf{z} \\
&= \alpha_{\omega}(\mathbf{x}) \int_{\omega-\boldsymbol{\mu}_{\omega}(\mathbf{x})} (\mathbf{y} + \boldsymbol{\mu}_{\omega}(\mathbf{x})) \frac{e^{-\frac{1}{2}\mathbf{y}^T \Sigma_{\omega}^{-1}\mathbf{y}}}{(2\pi)^{K/2} |\det(\Sigma_{\omega})|^{1/2}} d\mathbf{y} \\
&= \alpha_{\omega}(\mathbf{x}) \left(\mathbf{e}_{\omega-\boldsymbol{\mu}_{\omega}(\mathbf{x})}^1 + \mathbf{e}_{\omega-\boldsymbol{\mu}_{\omega}(\mathbf{x})}^0 \boldsymbol{\mu}_{\omega}(\mathbf{x}) \right)
\end{aligned}$$

Second moment:

$$\begin{aligned}
\int \mathbf{z} \mathbf{z}^T p(\mathbf{z}|\mathbf{x}) d\mathbf{z} &= \alpha_{\omega}(\mathbf{x}) \int_{\omega} \mathbf{z} \mathbf{z}^T \phi(\mathbf{z}; \boldsymbol{\mu}_{\omega}(\mathbf{x}), \Sigma_{\omega}) d\mathbf{z} \\
&= \alpha_{\omega}(\mathbf{x}) \int_{\omega-\boldsymbol{\mu}_{\omega}(\mathbf{x})} (\mathbf{y} + \boldsymbol{\mu}_{\omega}(\mathbf{x})) (\mathbf{y} + \boldsymbol{\mu}_{\omega}(\mathbf{x}))^T \phi(\mathbf{z}; \mathbf{0}, \Sigma_{\omega}) d\mathbf{y} \\
&= \alpha_{\omega}(\mathbf{x}) \left(\mathbf{E}^2 + \boldsymbol{\mu}_{\omega}(\mathbf{x}) \mathbf{e}_{\omega}^1(\Sigma_{\omega})^T + \mathbf{e}_{\omega}^1(\Sigma_{\omega}) \boldsymbol{\mu}_{\omega}(\mathbf{x})^T + \boldsymbol{\mu}_{\omega}(\mathbf{x}) \boldsymbol{\mu}_{\omega}(\mathbf{x})^T \mathbf{e}_{\omega-\boldsymbol{\mu}_{\omega}(\mathbf{x})}^0(\Sigma_{\omega}) \right)
\end{aligned}$$

□

G Proof of EM-step

We now derive the expectation maximization steps for a piecewise affine and continuous DGN.

G.1 E-step derivation

$$E_{\mathbf{z}|\mathbf{x}}[(A_{\omega}\mathbf{z} + B_{\omega})\mathbb{1}_{\omega}] = A m_{\omega}^1 + B e_{\omega}^0 \quad (20)$$

$$E_{\mathbf{z}|\mathbf{x}}[\mathbf{z}^T A_{\omega}^T A_{\omega} \mathbf{z} \mathbb{1}_{\omega}] = \text{Tr}(A_{\omega}^T A_{\omega} m_{\omega}^2) \quad (21)$$

$$\begin{aligned}
E_{Z|X} [\log(p_{X|Z}(\mathbf{x}|\mathbf{z}) p_Z(\mathbf{z}))] &= E_{Z|X} \left[\log \left(\frac{e^{-\frac{1}{2}(\mathbf{x}-g(\mathbf{z}))^T \Sigma_{\mathbf{x}}^{-1}(\mathbf{x}-g(\mathbf{z}))}}{(2\pi)^{D/2} \sqrt{|\det(\Sigma_{\mathbf{x})|}} \frac{e^{-\frac{1}{2}\mathbf{z}^T \Sigma_{\mathbf{z}}^{-1}\mathbf{z}}}{(2\pi)^{S/2} \sqrt{|\det(\Sigma_{\mathbf{z})|}}} \right) \right] \\
&= -\log \left((2\pi)^{(S+D)/2} \sqrt{|\det(\Sigma_{\mathbf{z}})|} \sqrt{|\det(\Sigma_{\mathbf{x}})|} \right) - \frac{1}{2} E_{Z|X} \left[(\mathbf{x} - g(\mathbf{z}))^T \Sigma_{\mathbf{x}}^{-1} (\mathbf{x} - g(\mathbf{z})) + \mathbf{z}^T \Sigma_{\mathbf{z}}^{-1} \mathbf{z} \right] \\
&= -\log \left((2\pi)^{(S+D)/2} \sqrt{|\det(\Sigma_{\mathbf{z}})|} \sqrt{|\det(\Sigma_{\mathbf{x}})|} \right) \\
&\quad - \frac{1}{2} \left(\mathbf{x}^T \Sigma_{\mathbf{x}}^{-1} \mathbf{x} + E_{Z|X} \left[-2\mathbf{x}^T \Sigma_{\mathbf{x}}^{-1} g(\mathbf{z}) + g(\mathbf{z})^T \Sigma_{\mathbf{x}}^{-1} g(\mathbf{z}) + \mathbf{z}^T \Sigma_{\mathbf{z}}^{-1} \mathbf{z} \right] \right) \\
&= -\log \left((2\pi)^{(S+D)/2} \sqrt{|\det(\Sigma_{\mathbf{z}})|} \sqrt{|\det(\Sigma_{\mathbf{x}})|} \right) - \frac{1}{2} \left(\mathbf{x}^T \Sigma_{\mathbf{x}}^{-1} \mathbf{x} + \text{Tr}(E_{Z|X}[\mathbf{z} \mathbf{z}^T \Sigma_{\mathbf{z}}^{-1}]) \right. \\
&\quad \left. + E_{Z|X} \left[-2\mathbf{x}^T \Sigma_{\mathbf{x}}^{-1} g(\mathbf{z}) + g(\mathbf{z})^T \Sigma_{\mathbf{x}}^{-1} g(\mathbf{z}) \right] \right) \\
&= -\log \left((2\pi)^{(S+D)/2} \sqrt{|\det(\Sigma_{\mathbf{z}})|} \sqrt{|\det(\Sigma_{\mathbf{x}})|} \right) - \frac{1}{2} \left(\mathbf{x}^T \Sigma_{\mathbf{x}}^{-1} \mathbf{x} - 2\mathbf{x}^T \Sigma_{\mathbf{x}}^{-1} \left(\sum_{\omega} \mathbf{A}_{\omega} e_{\omega}^1(\mathbf{x}) + \mathbf{b}_{\omega} e_{\omega}^0(\mathbf{x}) \right) \right. \\
&\quad \left. + \sum_{\omega} e_{\omega}^0 \mathbf{b}_{\omega}^T \Sigma_{\mathbf{x}}^{-1} \mathbf{b}_{\omega} + \text{Tr}(\mathbf{A}_{\omega}^T \Sigma_{\mathbf{x}}^{-1} \mathbf{A}_{\omega} \mathbf{E}_{\omega}^2(\mathbf{x})) + 2(\mathbf{A}_{\omega} \mathbf{m}_{\omega}^1(\mathbf{x}))^T \Sigma_{\mathbf{x}}^{-1} \mathbf{b}_{\omega} + \text{Tr}(\Sigma_{\mathbf{z}}^{-1} \mathbf{E}^2(\mathbf{x})) \right)
\end{aligned}$$

G.2 Proof of M step

Let first introduce some notations:

$$\begin{aligned} \mathbf{A}_\omega^{L \rightarrow i} &\triangleq (\mathbf{A}_\omega^{L \rightarrow i})^T \text{ (back-propagation matrix to layer } i), \\ r_\omega^\ell(\mathbf{x}) &\triangleq \left(\mathbf{x} e_\omega^0(\mathbf{x}) - \left(\mathbf{A}_\omega \mathbf{e}_\omega^1(\mathbf{x}) + \sum_{i \neq \ell} m_\omega^0(\mathbf{x}) \mathbf{A}_\omega^{i+1 \rightarrow L} \mathbf{D}_\omega^i \mathbf{v}^i \right) \right) \quad \text{(expected residual without } \mathbf{v}^\ell) \\ \hat{\mathbf{z}}_\omega^\ell(\mathbf{x}) &\triangleq \mathbf{D}_\omega^{\ell-1} (\mathbf{A}_\omega^{1 \rightarrow \ell-1} m_\omega^1(\mathbf{x}) + \mathbf{b}_\omega^{1 \rightarrow \ell-1} e_\omega^0) \quad \text{(expected feature map of layer } \ell) \end{aligned}$$

we can now provide the analytical forms of the M step for each of the learnable parameters:

$$\Sigma_{\mathbf{x}}^* = \frac{1}{N} \sum_{\mathbf{x}} \left(\mathbf{x} \mathbf{x}^T + \sum_{\omega} \mathbf{b}_\omega (\mathbf{b}_\omega m_\omega^0(\mathbf{x}) + 2 \mathbf{A}_\omega \mathbf{e}_\omega^1(\mathbf{x}))^T - 2 \mathbf{x} (\hat{\mathbf{z}}_\omega^L(\mathbf{x}))^T + \mathbf{A}_\omega \mathbf{E}_\omega^2(\mathbf{x}) \mathbf{A}_\omega^T \right), \quad (22)$$

$$\mathbf{v}^{\ell*} = \left(\sum_{\mathbf{x}} \sum_{\omega} \mathbf{D}_\omega^\ell \mathbf{A}_\omega^{L \rightarrow \ell+1} \Sigma_{\mathbf{x}}^{-1} \mathbf{A}_\omega^{\ell+1 \rightarrow L} \mathbf{D}_\omega^\ell \right)^{-1} \left(\sum_{\mathbf{x}} \sum_{\omega \in \Omega} \underbrace{\mathbf{D}_\omega^\ell \mathbf{A}_\omega^{L \rightarrow \ell+1} \Sigma_{\mathbf{x}}^{-1} r_\omega^\ell(\mathbf{x})}_{\text{residual back-propagated to layer } \ell} \right), \quad (23)$$

$$\text{vect}(\mathbf{W}^{\ell*}) = U_\omega^{-1} \text{vect} \left(\underbrace{\sum_{\mathbf{x}} \sum_{\omega} \mathbf{D}_\omega^\ell \mathbf{A}_\omega^{L \rightarrow \ell+1} \Sigma_{\mathbf{x}}^{-1} \left(\mathbf{x} - \sum_{i=\ell}^L \mathbf{A}_\omega^{i+1 \rightarrow L} \mathbf{D}_\omega^i \mathbf{v}^i \right)}_{\text{residual back-propagated to layer } \ell} (\hat{\mathbf{z}}_\omega^\ell(\mathbf{x}))^T \right), \quad (24)$$

we provide detailed derivations below.

G.2.1 Update of the bias parameter

Recall from (2) that $\mathbf{b}_\omega = \mathbf{v}^L + \sum_{i=1}^{L-1} \mathbf{W}^L \mathbf{D}_\omega^{L-1} \mathbf{W}^{L-1} \dots \mathbf{D}_\omega^i \mathbf{v}^i$, we can thus rewrite the loss as

$$\begin{aligned} L(\mathbf{v}^\ell) &= -\frac{1}{2} \log \left((2\pi)^{S+D} |\det(\Sigma_{\mathbf{x}})| |\det(\Sigma_{\mathbf{z}})| \right) - \frac{1}{2} \left(\mathbf{x}^T \Sigma_{\mathbf{x}}^{-1} \mathbf{x} - 2 \mathbf{x}^T \Sigma_{\mathbf{x}}^{-1} \left(\sum_{\omega} \mathbf{A}_\omega \mathbf{m}_\omega^1(\mathbf{x}) + \mathbf{b}_\omega m_\omega^0(\mathbf{x}) \right) \right. \\ &\quad \left. + \sum_{\omega} m_\omega^0 \mathbf{b}_\omega^T \Sigma_{\mathbf{x}}^{-1} \mathbf{b}_\omega + \text{Tr}(\mathbf{A}_\omega^T \Sigma_{\mathbf{x}}^{-1} \mathbf{A}_\omega \mathbf{M}_\omega^2(\mathbf{x})) + 2(\mathbf{A}_\omega \mathbf{m}_\omega^1(\mathbf{x}))^T \Sigma_{\mathbf{x}}^{-1} \mathbf{b}_\omega \right) - \frac{1}{2} \text{Tr}(\Sigma_{\mathbf{z}}^{-1} \mathbf{M}^2(\mathbf{x})) \\ &= -\frac{1}{2} \left(-2 \mathbf{x}^T \Sigma_{\mathbf{x}}^{-1} \left(\sum_{\omega} \mathbf{b}_\omega e_\omega^0(\mathbf{x}) \right) + \sum_{\omega} e_\omega^0 \mathbf{b}_\omega^T \Sigma_{\mathbf{x}}^{-1} \mathbf{b}_\omega + 2 \sum_{\omega} (\mathbf{A}_\omega \mathbf{m}_\omega^1(\mathbf{x}))^T \Sigma_{\mathbf{x}}^{-1} \mathbf{b}_\omega \right) + cst \\ &= -\frac{1}{2} \sum_{\omega} \left(-2 \mathbf{x}^T \Sigma_{\mathbf{x}}^{-1} (\mathbf{A}_\omega^{\ell+1 \rightarrow L} \mathbf{D}_\omega^\ell \mathbf{v}^\ell e_\omega^0(\mathbf{x})) + e_\omega^0 (\mathbf{A}_\omega^{\ell+1 \rightarrow L} \mathbf{D}_\omega^\ell \mathbf{v}^\ell)^T \Sigma_{\mathbf{x}}^{-1} (\mathbf{A}_\omega^{\ell+1 \rightarrow L} \mathbf{D}_\omega^\ell \mathbf{v}^\ell) \right. \\ &\quad \left. + 2 e_\omega^0(\mathbf{x}) \left(\sum_{i \neq \ell} \mathbf{A}_\omega^{i+1 \rightarrow L} \mathbf{D}_\omega^i \mathbf{v}^i \right)^T \Sigma_{\mathbf{x}}^{-1} (\mathbf{A}_\omega^{\ell+1 \rightarrow L} \mathbf{D}_\omega^\ell \mathbf{v}^\ell) + 2((m_\omega^1(\mathbf{x}))^T (\mathbf{A}_\omega)^T \Sigma_{\mathbf{x}}^{-1} \mathbf{A}_\omega^{\ell+1 \rightarrow L} \mathbf{D}_\omega^\ell \mathbf{v}^\ell) \right) + cst \\ &= -\frac{1}{2} \sum_{\omega} \left(e_\omega^0 (\mathbf{A}_\omega^{\ell+1 \rightarrow L} \mathbf{D}_\omega^\ell \mathbf{v}^\ell)^T \Sigma_{\mathbf{x}}^{-1} (\mathbf{A}_\omega^{\ell+1 \rightarrow L} \mathbf{D}_\omega^\ell \mathbf{v}^\ell) \right. \end{aligned} \quad (A)$$

$$\left. + 2(e_\omega^0(\mathbf{x}) \left(\sum_{i \neq \ell} \mathbf{A}_\omega^{i+1 \rightarrow L} \mathbf{D}_\omega^i \mathbf{v}^i - \mathbf{x} \right) + \mathbf{A}_\omega \mathbf{e}_\omega^1(\mathbf{x}))^T \Sigma_{\mathbf{x}}^{-1} (\mathbf{A}_\omega^{\ell+1 \rightarrow L} \mathbf{D}_\omega^\ell \mathbf{v}^\ell) \right) + cst \quad (B)$$

$$\begin{aligned}
\Rightarrow \partial L(\mathbf{v}^\ell) &= -\frac{1}{2} \sum_{\omega} \left[-e_{\omega}^0(\mathbf{x}) 2\mathbf{D}_{\omega}^{\ell} \mathbf{A}_{\omega}^{L \rightarrow \ell+1} \boldsymbol{\Sigma}_{\mathbf{x}}^{-1} \mathbf{A}_{\omega}^{\ell+1 \rightarrow L} \mathbf{D}_{\omega}^{\ell} \mathbf{v}^{\ell} \right. \\
&\quad \left. + 2(\mathbf{A}_{\omega}^{\ell+1 \rightarrow L} \mathbf{D}_{\omega}^{\ell})^T \boldsymbol{\Sigma}_{\mathbf{x}}^{-1} \left(e_{\omega}^0(\mathbf{x}) \left(\sum_{i \neq \ell} \mathbf{A}_{\omega}^{i+1 \rightarrow L} \mathbf{D}_{\omega}^i \mathbf{v}^i - \mathbf{x} \right) + \mathbf{A}_{\omega} \mathbf{e}_{\omega}^1(\mathbf{x}) \right) \right] \\
\Rightarrow \mathbf{v}^{\ell} &= \left(\sum_{\mathbf{x}} \sum_{\omega} e_{\omega}^0(\mathbf{x}) \mathbf{D}_{\omega}^{\ell} \mathbf{A}_{\omega}^{L \rightarrow \ell+1} \boldsymbol{\Sigma}_{\mathbf{x}}^{-1} \mathbf{A}_{\omega}^{\ell+1 \rightarrow L} \mathbf{D}_{\omega}^{\ell} \right)^{-1} \\
&\quad \times \sum_{\mathbf{x}} \sum_{\omega \in \Omega} \mathbf{D}_{\omega}^{\ell} \mathbf{A}_{\omega}^{L \rightarrow \ell+1} \boldsymbol{\Sigma}_{\mathbf{x}}^{-1} \left(\mathbf{x} e_{\omega}^0(\mathbf{x}) - \left(\mathbf{A}_{\omega} m_{\omega}^1(\mathbf{x}) + \sum_{i \neq \ell} m_{\omega}^0(\mathbf{x}) \mathbf{A}_{\omega}^{i+1 \rightarrow L} \mathbf{D}_{\omega}^i \mathbf{v}^i \right) \right)
\end{aligned}$$

as

$$\begin{aligned}
(A) &= e_{\omega}^0(\mathbf{x}) (\mathbf{A}_{\omega}^{\ell+1 \rightarrow L} \mathbf{D}_{\omega}^{\ell} \mathbf{v}^{\ell})^T \boldsymbol{\Sigma}_{\mathbf{x}}^{-1} (\mathbf{A}_{\omega}^{\ell+1 \rightarrow L} \mathbf{D}_{\omega}^{\ell} \mathbf{v}^{\ell}) \\
\Rightarrow \partial(A) &= e_{\omega}^0(\mathbf{x}) 2\mathbf{D}_{\omega}^{\ell} \mathbf{A}_{\omega}^{L \rightarrow \ell+1} \boldsymbol{\Sigma}_{\mathbf{x}}^{-1} \mathbf{A}_{\omega}^{\ell+1 \rightarrow L} \mathbf{D}_{\omega}^{\ell} \mathbf{v}^{\ell} \\
(B) &= 2 \left[\left(e_{\omega}^0(\mathbf{x}) \left(\sum_{i \neq \ell} \mathbf{A}_{\omega}^{i+1 \rightarrow L} \mathbf{D}_{\omega}^i \mathbf{v}^i - \mathbf{x} \right) + \mathbf{A}_{\omega} \mathbf{e}_{\omega}^1(\mathbf{x}) \right)^T \boldsymbol{\Sigma}_{\mathbf{x}}^{-1} (\mathbf{A}_{\omega}^{\ell+1 \rightarrow L} \mathbf{D}_{\omega}^{\ell} \mathbf{v}^{\ell}) \right] + cst \\
\Rightarrow \partial(B) &= (\mathbf{A}_{\omega}^{\ell+1 \rightarrow L} \mathbf{D}_{\omega}^{\ell})^T \boldsymbol{\Sigma}_{\mathbf{x}}^{-1} \left(e_{\omega}^0(\mathbf{x}) \left(\sum_{i \neq \ell} \mathbf{A}_{\omega}^{i+1 \rightarrow L} \mathbf{D}_{\omega}^i \mathbf{v}^i - \mathbf{x} \right) + \mathbf{A}_{\omega} \mathbf{e}_{\omega}^1(\mathbf{x}) \right)
\end{aligned}$$

G.2.2 Update of the slope parameter

We can thus rewrite the loss as

$$\begin{aligned}
L(\mathbf{v}^{\ell}) &= -\frac{1}{2} \log \left((2\pi)^{S+D} |\det(\boldsymbol{\Sigma}_{\mathbf{x}})| |\det(\boldsymbol{\Sigma}_{\mathbf{z}})| \right) - \frac{1}{2} \left(\mathbf{x}^T \boldsymbol{\Sigma}_{\mathbf{x}}^{-1} \mathbf{x} - 2\mathbf{x}^T \boldsymbol{\Sigma}_{\mathbf{x}}^{-1} \left(\sum_{\omega} \mathbf{A}_{\omega} m_{\omega}^1(\mathbf{x}) + \mathbf{b}_{\omega} m_{\omega}^0(\mathbf{x}) \right) \right. \\
&\quad \left. + \sum_{\omega} m_{\omega}^0 \mathbf{b}_{\omega}^T \boldsymbol{\Sigma}_{\mathbf{x}}^{-1} \mathbf{b}_{\omega} + \text{Tr}(\mathbf{A}_{\omega}^T \boldsymbol{\Sigma}_{\mathbf{x}}^{-1} \mathbf{A}_{\omega} M_{\omega}^2(\mathbf{x})) + 2(\mathbf{A}_{\omega} m_{\omega}^1(\mathbf{x}))^T \boldsymbol{\Sigma}_{\mathbf{x}}^{-1} \mathbf{b}_{\omega} \right) - \frac{1}{2} \text{Tr}(\boldsymbol{\Sigma}_{\mathbf{z}}^{-1} M^2(\mathbf{x})) \\
&= \mathbf{x}^T \boldsymbol{\Sigma}_{\mathbf{x}}^{-1} \left(\sum_{\omega} \mathbf{A}_{\omega} \mathbf{e}_{\omega}^1(\mathbf{x}) + \mathbf{b}_{\omega} e_{\omega}^0(\mathbf{x}) \right) - \frac{1}{2} \sum_{\omega} e_{\omega}^0 \mathbf{b}_{\omega}^T \boldsymbol{\Sigma}_{\mathbf{x}}^{-1} \mathbf{b}_{\omega} - \frac{1}{2} \sum_{\omega} \text{Tr}(\mathbf{A}_{\omega}^T \boldsymbol{\Sigma}_{\mathbf{x}}^{-1} \mathbf{A}_{\omega} \mathbf{E}_{\omega}^2(\mathbf{x})) \\
&\quad - \sum_{\omega} (\mathbf{A}_{\omega} m_{\omega}^1(\mathbf{x}))^T \boldsymbol{\Sigma}_{\mathbf{x}}^{-1} \mathbf{b}_{\omega}
\end{aligned}$$

Notice that we can rewrite $\mathbf{b}_{\omega} = \mathbf{A}^{\ell+1 \rightarrow L} \mathbf{D}_{\omega}^{\ell} \mathbf{W}^{\ell} \mathbf{D}_{\omega}^{\ell-1} \mathbf{b}_{\omega}^{1 \rightarrow \ell-1} + \sum_{i=\ell}^L \mathbf{A}_{\omega}^{i+1 \rightarrow L} \mathbf{D}_{\omega}^i \mathbf{v}^i$ and $\mathbf{A}_{\omega} = \mathbf{A}_{\omega}^{\ell+1 \rightarrow L} \mathbf{D}_{\omega}^{\ell} \mathbf{W}^{\ell} \mathbf{D}_{\omega}^{\ell-1} \mathbf{A}^{1 \rightarrow \ell-1}$ and thus we obtain:

$$\begin{aligned}
L(\mathbf{v}^{\ell}) &= \sum_{\omega} \mathbf{x}^T \boldsymbol{\Sigma}_{\mathbf{x}}^{-1} \mathbf{A}^{\ell+1 \rightarrow L} \mathbf{D}_{\omega}^{\ell} \mathbf{W}^{\ell} \mathbf{D}_{\omega}^{\ell-1} \left(\mathbf{A}^{1 \rightarrow \ell-1} \mathbf{e}_{\omega}^1(\mathbf{x}) + \mathbf{b}_{\omega}^{1 \rightarrow \ell-1} e_{\omega}^0(\mathbf{x}) \right) \\
&\quad - \frac{1}{2} \sum_{\omega} e_{\omega}^0 (\mathbf{A}^{\ell+1 \rightarrow L} \mathbf{D}_{\omega}^{\ell} \mathbf{W}^{\ell} \mathbf{D}_{\omega}^{\ell-1} \mathbf{b}_{\omega}^{1 \rightarrow \ell-1})^T \boldsymbol{\Sigma}_{\mathbf{x}}^{-1} (\mathbf{A}^{\ell+1 \rightarrow L} \mathbf{D}_{\omega}^{\ell} \mathbf{W}^{\ell} \mathbf{D}_{\omega}^{\ell-1} \mathbf{b}_{\omega}^{1 \rightarrow \ell-1})
\end{aligned}$$

$$\begin{aligned}
& - \sum_{\omega} e_{\omega}^0 (A^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} b_{\omega}^{1 \rightarrow \ell-1})^T \Sigma_{\mathbf{x}}^{-1} \left(\sum_{i=\ell}^L A_{\omega}^{i+1 \rightarrow L} D_{\omega}^i v^i \right) \\
& - \frac{1}{2} \sum_{\omega} \text{Tr}((A_{\omega}^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} A^{1 \rightarrow \ell-1})^T \Sigma_{\mathbf{x}}^{-1} (A_{\omega}^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} A^{1 \rightarrow \ell-1}) E_{\omega}^2(\mathbf{x})) \\
& - \sum_{\omega} (A_{\omega}^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} A^{1 \rightarrow \ell-1} m_{\omega}^1(\mathbf{x}))^T \Sigma_{\mathbf{x}}^{-1} (A^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} b_{\omega}^{1 \rightarrow \ell-1}) \\
& - \sum_{\omega} (A_{\omega}^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} A^{1 \rightarrow \ell-1} m_{\omega}^1(\mathbf{x}))^T \Sigma_{\mathbf{x}}^{-1} \left(\sum_{i=\ell}^L A_{\omega}^{i+1 \rightarrow L} D_{\omega}^i v^i \right) + cst \\
= & \sum_{\omega} \mathbf{x}^T \Sigma_{\mathbf{x}}^{-1} A^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} \left(A^{1 \rightarrow \ell-1} e_{\omega}^1(\mathbf{x}) + b_{\omega}^{1 \rightarrow \ell-1} e_{\omega}^0(\mathbf{x}) \right) \tag{A}
\end{aligned}$$

$$- \frac{1}{2} \sum_{\omega} e_{\omega}^0 (A^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} b_{\omega}^{1 \rightarrow \ell-1})^T \Sigma_{\mathbf{x}}^{-1} (A^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} b_{\omega}^{1 \rightarrow \ell-1}) \tag{B}$$

$$- \sum_{\omega} \left(A^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} (A^{1 \rightarrow \ell-1} e_{\omega}^1(\mathbf{x}) + b_{\omega}^{1 \rightarrow \ell-1} e_{\omega}^0(\mathbf{x})) \right)^T \Sigma_{\mathbf{x}}^{-1} \left(\sum_{i=\ell}^L A_{\omega}^{i+1 \rightarrow L} D_{\omega}^i v^i \right) \tag{C}$$

$$- \frac{1}{2} \sum_{\omega} \text{Tr}((A_{\omega}^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} A^{1 \rightarrow \ell-1})^T \Sigma_{\mathbf{x}}^{-1} (A_{\omega}^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} A^{1 \rightarrow \ell-1}) E_{\omega}^2(\mathbf{x})) \tag{D}$$

$$- \sum_{\omega} (A_{\omega}^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} A^{1 \rightarrow \ell-1} m_{\omega}^1(\mathbf{x}))^T \Sigma_{\mathbf{x}}^{-1} (A^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} b_{\omega}^{1 \rightarrow \ell-1}) + cst \tag{E}$$

$$\begin{aligned}
A &= \sum_{\omega} \mathbf{x}^T \Sigma_{\mathbf{x}}^{-1} A^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} \left(A^{1 \rightarrow \ell-1} e_{\omega}^1(\mathbf{x}) + b_{\omega}^{1 \rightarrow \ell-1} e_{\omega}^0(\mathbf{x}) \right) \\
\Rightarrow \partial A &= \sum_{\omega} D_{\omega}^{\ell} A_{\omega}^{L \rightarrow \ell+1} \Sigma_{\mathbf{x}}^{-1} \mathbf{x} \left(D_{\omega}^{\ell-1} (A^{1 \rightarrow \ell-1} e_{\omega}^1(\mathbf{x}) + b_{\omega}^{1 \rightarrow \ell-1} e_{\omega}^0(\mathbf{x})) \right)^T
\end{aligned}$$

$$\begin{aligned}
B &= - \frac{1}{2} \sum_{\omega} e_{\omega}^0(\mathbf{x}) (A^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} b_{\omega}^{1 \rightarrow \ell-1})^T \Sigma_{\mathbf{x}}^{-1} (A^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} b_{\omega}^{1 \rightarrow \ell-1}) \\
&= - \frac{1}{2} \sum_{\omega} e_{\omega}^0(\mathbf{x}) (D_{\omega}^{\ell-1} b_{\omega}^{1 \rightarrow \ell-1})^T (W^{\ell})^T (A^{\ell+1 \rightarrow L} D_{\omega}^{\ell})^T \Sigma_{\mathbf{x}}^{-1} (A^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} b_{\omega}^{1 \rightarrow \ell-1}) \\
&= - \frac{1}{2} \sum_{\omega} e_{\omega}^0(\mathbf{x}) \text{Tr}((W^{\ell})^T (A^{\ell+1 \rightarrow L} D_{\omega}^{\ell})^T \Sigma_{\mathbf{x}}^{-1} A^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} b_{\omega}^{1 \rightarrow \ell-1} (D_{\omega}^{\ell-1} b_{\omega}^{1 \rightarrow \ell-1})^T) \\
\Rightarrow \partial B &= - \sum_{\omega} e_{\omega}^0(\mathbf{x}) D_{\omega}^{\ell} A_{\omega}^{L \rightarrow \ell+1} \Sigma_{\mathbf{x}}^{-1} A^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} b_{\omega}^{1 \rightarrow \ell-1} (b_{\omega}^{1 \rightarrow \ell-1})^T D_{\omega}^{\ell-1}
\end{aligned}$$

$$C = - \sum_{\omega} \left(A^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} (A^{1 \rightarrow \ell-1} e_{\omega}^1(\mathbf{x}) + b_{\omega}^{1 \rightarrow \ell-1} e_{\omega}^0(\mathbf{x})) \right)^T \Sigma_{\mathbf{x}}^{-1} \left(\sum_{i=\ell}^L A_{\omega}^{i+1 \rightarrow L} D_{\omega}^i v^i \right)$$

$$\begin{aligned}
&= - \sum_{\omega} (D_{\omega}^{\ell-1} (A^{1 \rightarrow \ell-1} e_{\omega}^1(x) + b_{\omega}^{1 \rightarrow \ell-1} e_{\omega}^0(x)))^T (W^{\ell})^T (A^{\ell+1 \rightarrow L} D_{\omega}^{\ell})^T \Sigma_{\mathbf{x}}^{-1} \left(\sum_{i=\ell}^L A_{\omega}^{i+1 \rightarrow L} D_{\omega}^i v^i \right) \\
\Rightarrow \partial C &= - \sum_{\omega} D_{\omega}^{\ell} A^{L \rightarrow \ell+1} \Sigma_{\mathbf{x}}^{-1} \left(\sum_{i=\ell}^L A_{\omega}^{i+1 \rightarrow L} D_{\omega}^i v^i \right) (D_{\omega}^{\ell-1} (A^{1 \rightarrow \ell-1} e_{\omega}^1(x) + b_{\omega}^{1 \rightarrow \ell-1} e_{\omega}^0(x)))^T
\end{aligned}$$

$$\begin{aligned}
D &= - \frac{1}{2} \sum_{\omega} \text{Tr}((A_{\omega}^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} A^{1 \rightarrow \ell-1})^T \Sigma_{\mathbf{x}}^{-1} A_{\omega}^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} A^{1 \rightarrow \ell-1} E_{\omega}^2(x)) \\
&= - \frac{1}{2} \sum_{\omega} \text{Tr}((W^{\ell})^T (A_{\omega}^{\ell+1 \rightarrow L} D_{\omega}^{\ell})^T \Sigma_{\mathbf{x}}^{-1} A_{\omega}^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} A^{1 \rightarrow \ell-1} E_{\omega}^2(x) (D_{\omega}^{\ell-1} A_{\omega}^{1 \rightarrow \ell-1})^T) \\
\Rightarrow \partial D &= - \sum_{\omega} D_{\omega}^{\ell} A^{L \rightarrow \ell+1} \Sigma_{\mathbf{x}}^{-1} A_{\omega}^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} A^{1 \rightarrow \ell-1} E_{\omega}^2(x) A_{\omega}^{\ell-1 \rightarrow 1} D_{\omega}^{\ell-1}
\end{aligned}$$

$$\begin{aligned}
E &= - \sum_{\omega} (A_{\omega}^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} A^{1 \rightarrow \ell-1} m_{\omega}^1(x))^T \Sigma_{\mathbf{x}}^{-1} (A_{\omega}^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} b_{\omega}^{1 \rightarrow \ell-1}) \\
&= - \sum_{\omega} \text{Tr}((W^{\ell})^T (A_{\omega}^{\ell+1 \rightarrow L} D_{\omega}^{\ell})^T \Sigma_{\mathbf{x}}^{-1} A_{\omega}^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} b_{\omega}^{1 \rightarrow \ell-1} (D_{\omega}^{\ell-1} A^{1 \rightarrow \ell-1} m_{\omega}^1(x))^T) \\
\Rightarrow \partial E &= - \sum_{\omega} D_{\omega}^{\ell} A^{L \rightarrow \ell+1} \Sigma_{\mathbf{x}}^{-1} A_{\omega}^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} (D_{\omega}^{\ell-1} (b_{\omega}^{1 \rightarrow \ell-1} (m_{\omega}^1(x))^T A_{\omega}^{\ell-1 \rightarrow 1} + A_{\omega}^{1 \rightarrow \ell-1} m_{\omega}^1(x) (b_{\omega}^{1 \rightarrow \ell-1})^T) (D_{\omega}^{\ell-1})^T)
\end{aligned}$$

we can group B,D and E together as well as A and C. Now to solve this equal 0 we will need to consider the flatten version of W^{ℓ} which we denote by $w^{\ell} = \text{vect}(W^{\ell})$ leading to

$$\begin{aligned}
\partial L &= \sum_{\omega} D_{\omega}^{\ell} A_{\omega}^{L \rightarrow \ell+1} \Sigma_{\mathbf{x}}^{-1} \left(x - \sum_{i=\ell}^L A_{\omega}^{i+1 \rightarrow L} D_{\omega}^i v^i \right) (D_{\omega}^{\ell-1} (A^{1 \rightarrow \ell-1} e_{\omega}^1(x) + b_{\omega}^{1 \rightarrow \ell-1} e_{\omega}^0(x)))^T \\
&\quad - \sum_{\omega} D_{\omega}^{\ell} A^{L \rightarrow \ell+1} \Sigma_{\mathbf{x}}^{-1} A_{\omega}^{\ell+1 \rightarrow L} D_{\omega}^{\ell} W^{\ell} D_{\omega}^{\ell-1} \left(e_{\omega}^0(x) b_{\omega}^{1 \rightarrow \ell-1} (b_{\omega}^{1 \rightarrow \ell-1})^T + A_{\omega}^{1 \rightarrow \ell-1} E_{\omega}^2(x) A_{\omega}^{\ell-1 \rightarrow 1} \right. \\
&\quad \left. + b_{\omega}^{1 \rightarrow \ell-1} (m_{\omega}^1(x))^T A_{\omega}^{\ell-1 \rightarrow 1} + A_{\omega}^{1 \rightarrow \ell-1} m_{\omega}^1(x) (b_{\omega}^{1 \rightarrow \ell-1})^T \right) D_{\omega}^{\ell-1} \\
&= \sum_{\omega} P_{\omega}(x)^{\ell} - U_{\omega}^{\ell} W^{\ell} V_{\omega}^{\ell}(x) \\
\Rightarrow & \left(\sum_{\mathbf{x}} \sum_{\omega} U_{\omega}^{\ell} \otimes (V_{\omega}^{\ell}(x))^T \right) \text{vect}(W^{\ell}) = \sum_{\mathbf{x}} \sum_{\omega} \text{vect}(P_{\omega}^{\ell}(x)) \\
\Rightarrow \text{vect}(W^{\ell})^* &= \left(\sum_{\mathbf{x}} \sum_{\omega} U_{\omega}^{\ell} \otimes (V_{\omega}^{\ell}(x))^T \right)^{-1} \left(\sum_{\mathbf{x}} \sum_{\omega} \text{vect}(P_{\omega}^{\ell}(x)) \right)
\end{aligned}$$

H Regularization

We propose in this section a brief discussion on the impact of using a probabilistic prior on the weights of the GDN. In particular, it is clear that imposing a Gaussian prior with zero mean and isotropic covariance on the weights falls back in the log likelihood to impose a $l2$ regularization of the weights with parameter based on the covariance of the prior. If the prior is a Laplace distribution,

the log-likelihood will turn the prior into an l_1 regularization of the weights, again with regularization coefficient based on the prior covariance. Finally, in the case of uniform prior with finite support, the log likelihood will be equivalent to a weight clipping, a standard technique employed in DNs where the weights can not take values outside of a predefined range.

I Computational Complexity

The computational complexity of the method increases drastically with the latent space dimension, and the number of regions, and the number of faces per regions. Those last quantities are directly tied into the complexity (depth and width) of the DGNs. This complexity bottleneck comes from the need to search for all regions, and the need to decompose each region into simplices. As such, the EM learning is not yet suitable for large scale application, however based on the obtained analytical forms, it is possible to derive an approximation of the true form that would be more tractable while providing approximation error bounds as opposed to current methods.

J Additional Experiments

In this section we propose to complement the toy circle experiment from the main paper first we an additional $2d$ case and then with the MNIST dataset.

Wave

We propose here a simple example where the read data is as follows:

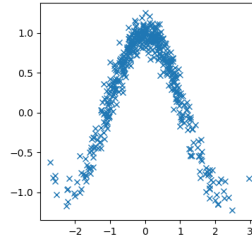


Figure 7: sample of noise data for the wave dataset

We train on this dataset the EM and VAE based learning with various learning rates and depict below the evolution of the NLL for all models, we also depict the samples after learning.

MNIST We now employ MNIST which consists of images of digits, and select the 4 class. Note that due to complexity overhead we maintain a univariate latent space of the GDN and employ a three layer DGN with 8 and 16 hidden units. We provide first the evolution of the NLL through learning for all the training methods and then sample images from the trained DGNs demonstrating how for small DGNs EM learning is able to learn a better data distribution and thus generated realistic samples as opposed to VAEs which need much longer training steps.

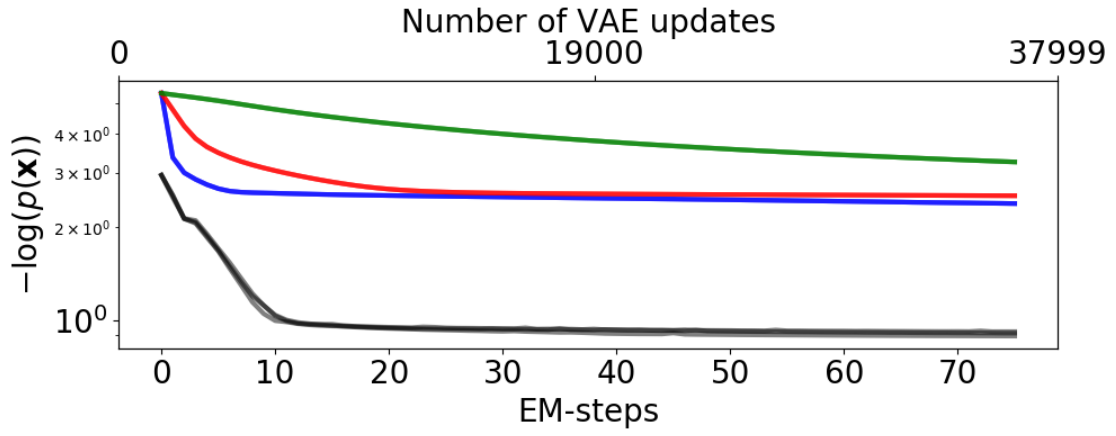


Figure 8: Depiction of the evolution of the NLL during training for the EM and VAE algorithms, we can see that despite the high number of training steps, VAEs are not yet able to correctly approximate the data distribution as opposed to EM training which benefits from much faster convergence. We also see how the VAEs tend to have a large KL divergence between the true posterior and the variational estimate due to this gap, we depict below samples from those models.

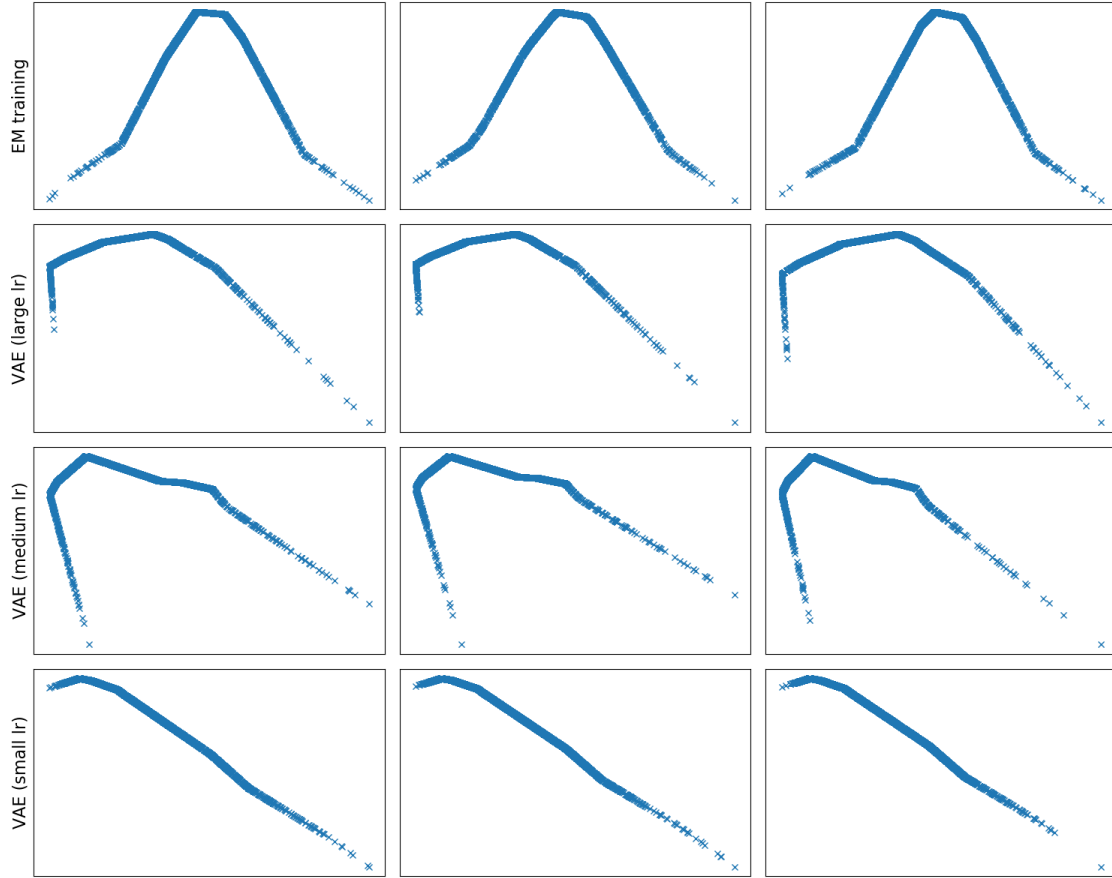


Figure 9: Samples from the various models trained on the wave dataset. We can see on **top** the result of EM training where each column represents a different run, the remaining three rows correspond to the VAE training. Again, EM demonstrates much faster convergence, for VAE to reach the actual data distribution, much more updates are needed.

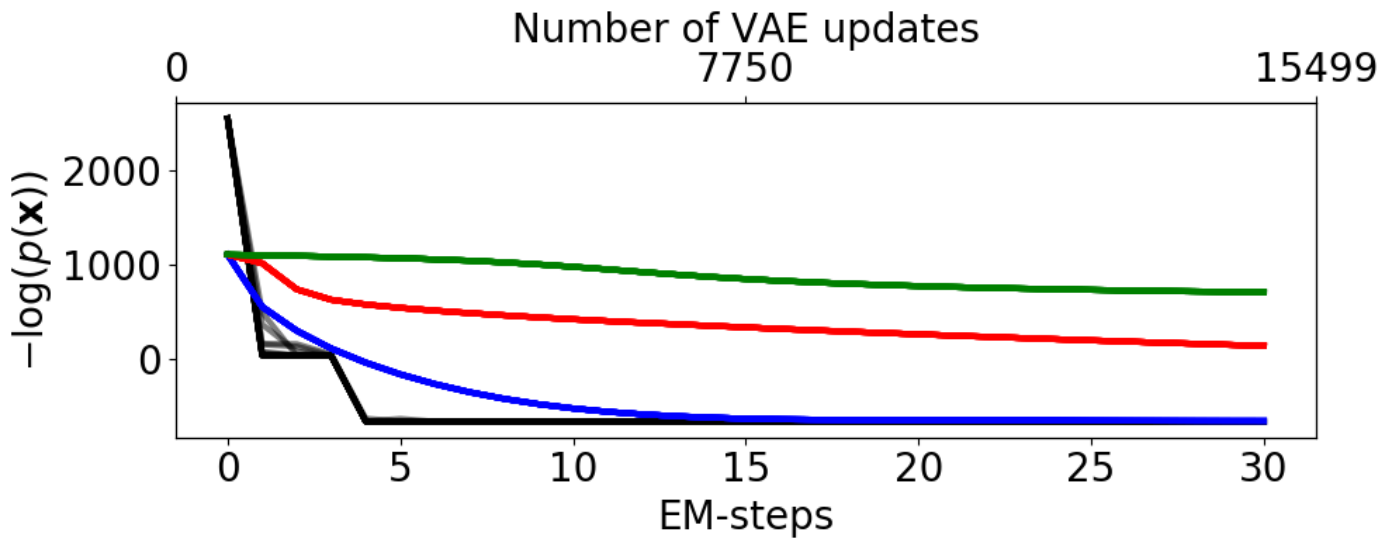


Figure 10: Evolution of the true data negative log-likelihood (in semilog-y plot on MNIST (class 4) for EM and VAE training for a small DGN as described above. The experiments are repeated multiple times, we can see how the learning rate is clearly impacting the learning significantly despite the use of Adam, and that even with the large learning rate, the EM learning is able to reach lower NLL, in fact the quality of the generated samples of the EM modes is much higher as shows below.

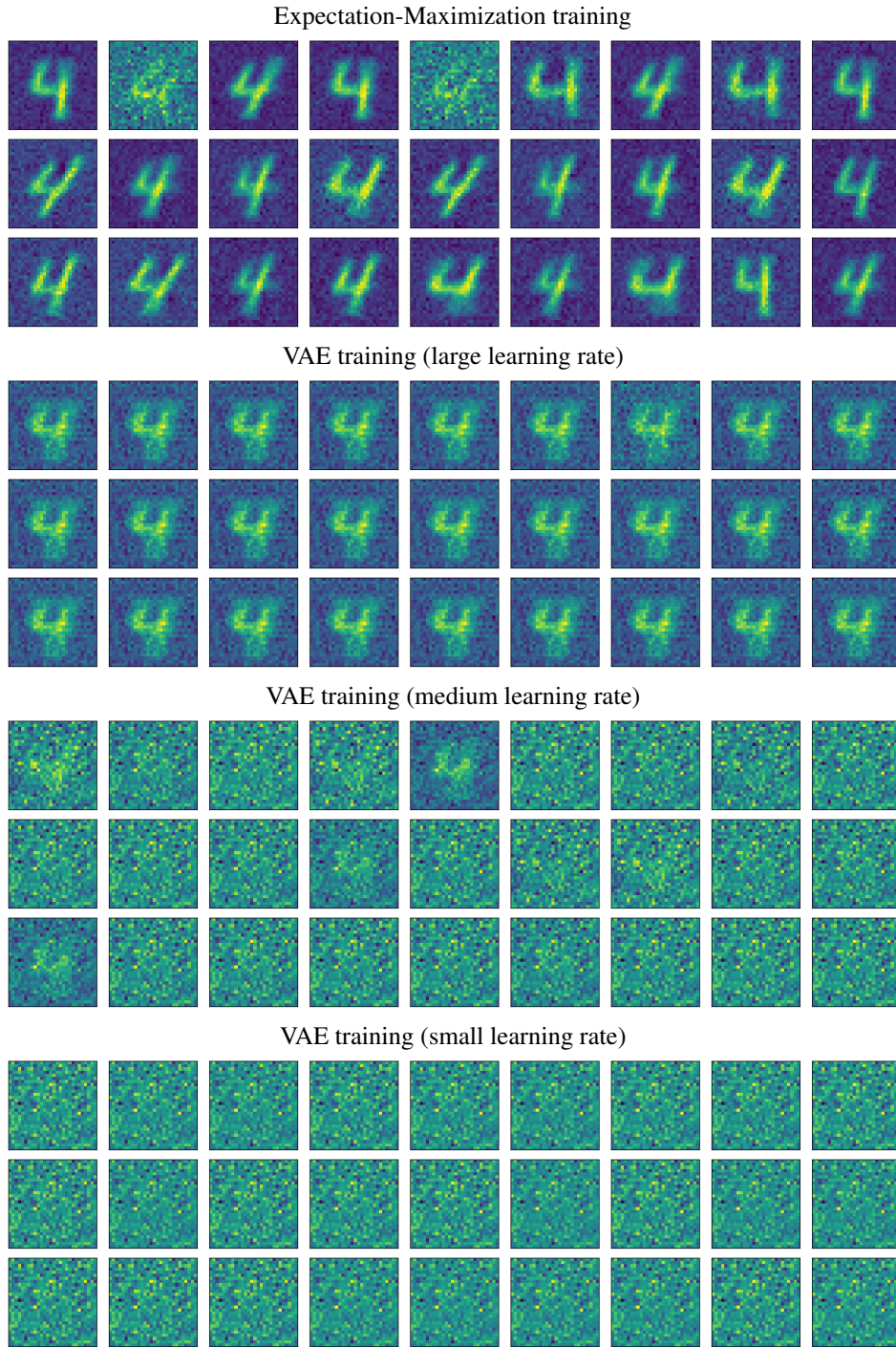


Figure 11: Random samples from trained DGNs with EM or VAEs on a MNIST experiment (with digit 4). We see the ability of EM training to produce realistic and diversified samples despite using a latent space dimension of 1 and a small generative network.