Whitening Convergence Rate of Coupling-based Normalizing Flows

Felix Draxler
Heidelberg University
felix.draxler@iwr.uni-heidelberg.de

Christoph Schnörr
Heidelberg University
schnoerr@math.uni-heidelberg.de

Ullrich Köthe
Heidelberg University
ullrich.koethe@iwr.uni-heidelberg.de

Abstract

Coupling-based normalizing flows (e.g. RealNVP) are a popular family of normalizing flow architectures that work surprisingly well in practice. This calls for theoretical understanding. Existing work shows that such flows *weakly* converge to arbitrary data distributions [1]. However, they make no statement about the stricter convergence criterion used in practice, the maximum likelihood loss. For the first time, we make a quantitative statement about this kind of convergence: We prove that all coupling-based normalizing flows perform whitening of the data distribution (i.e. diagonalize the covariance matrix) and derive corresponding convergence bounds that show a linear convergence rate in the depth of the flow. Numerical experiments demonstrate the implications of our theory and point at open questions.

1 Introduction

Normalizing flows [2, 3] are among the most promising approaches to *generative* machine learning and have already demonstrated convincing performance in a wide variety of practical applications, ranging from image analysis [4, 5, 6, 7, 8] to astrophysics [9], mechanical engineering [10], causality [11], computational biology [12] and medicine [13]. As the name suggests, normalizing flows represent complex data distributions as bijective transformations (also known as flows or *push-forwards*) of standard normal or other well-understood distributions.

In this paper, we focus on a theoretical underpinning of coupling-based normalizing flows, a particularly effective class of normalizing flows in terms of invertible neural networks. All of the above applications are actually implemented using coupling-based normalizing flows. Their central building blocks are *coupling layers*, which decompose the space into two subspaces called *active* and *passive* subspace (see Section 3). Only the active dimensions are transformed conditioned on the passive dimensions, which makes the mapping computationally easy to invert. In order to vary the assignment of dimensions to the active and passive subspaces, coupling layers are combined with preceding orthonormal transformation layers into *coupling blocks*. These blocks are arranged into deep networks such that the orthonormal transformations are sampled uniformly at random from the orthogonal matrices and the coupling layers are trained with the maximum likelihood objective, see Equation (2). Upon convergence of the training, the sequence of coupling blocks gradually transforms the probability density that generated the given training data, into a standard normal distribution and vice versa.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

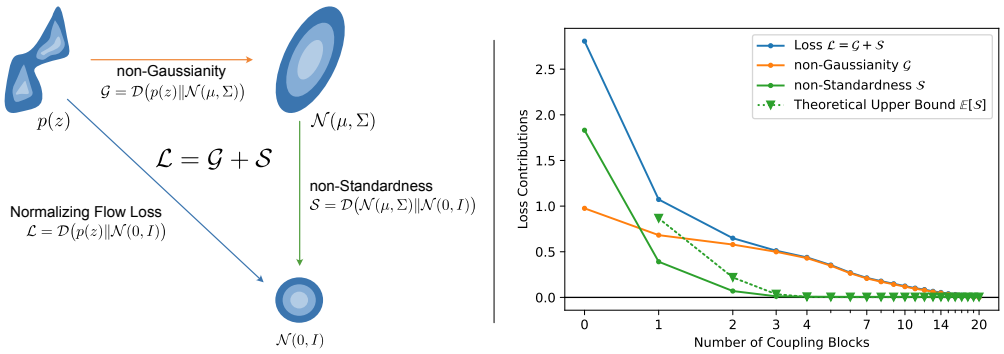

Figure 1: *(Left)* The Maximum Likelihood Loss $\mathcal{L}$ (blue) can be split into the *non-Gaussianity* $\mathcal{G}$ (orange) [25] and the *non-Standardness* $\mathcal{S}$ (green) of the latent code $z = f_\theta(x)$: $\mathcal{L} = \mathcal{G} + \mathcal{S}$ (Proposition 1). For the latter, we give explicit guarantees as one more coupling block is added in Theorems 1 and 2 and show a global convergence rate in Theorem 3. *(Right)* Typical fit of EMNIST digits by a standard affine coupling flow for various depths. Our theory (Theorem 1) upper bounds the average $\mathcal{S}$ for $L+1$ coupling blocks given a trained model with L coupling blocks (dotted green). We observe that our bound is predictive for how much end-to-end training reduces $\mathcal{S}$.

Since the resulting normalizing flows deviate significantly from *optimal* transport flows [14] and the bulk of the mathematical literature is focusing on optimal transport, an analysis tailored to coupling architectures is lacking. In a landmark paper, [1] proved that sufficiently large affine coupling flows weakly converge to arbitrary data densities. The notion of weak convergence is critical here, as *it does not imply convergence in maximum likelihood* [15, Remark 3]. Maximum likelihood (or, equivalently, the Kullback-Leibler (KL) divergence) is the loss that is actually used in practice. It can be used for gradient descent and it guarantees not only convergence in samples ("$x \sim q(x) \to x \sim p(x)$") but also in density estimates ("$q(x) \to p(x)$"). It is strong in the sense that the square root of the KL divergence upper bounds (up to a factor 2) the total variation metric, and hence also the Wasserstein metric if the underlying space is bounded [16]. Moreover, convergence under the KL divergence implies weak convergence which is fundamental for robust statistics [17].

We take a first step towards showing that coupling blocks also converge in terms of maximum likelihood. To the best of our knowledge, our paper presents for the first time a quantitative convergence analysis of coupling-based normalizing flows based on this strong notion of convergence.

Specifically, we make the following contributions towards this goal:

- We utilize that the loss of a normalizing flow can be decomposed into two parts (Figure 1): The divergence to the nearest Gaussian (*non-Gaussianity*) plus the divergence of that Gaussian to the standard normal (*non-Standardness*).
- The contribution of a single coupling layer on the non-Standardness is analyzed in terms of matrix operations (Schur complement and scaling).
- Explicit bounds for the non-Standardness after a single coupling block in expectation over all orthonormal transformations are derived.
- We use these results to prove that a sequence of coupling blocks whitens the data covariance and to derive linear convergence rates for this process.

Our results hold for all coupling architectures we are aware of (Appendix C), including: NICE [4], RealNVP [5], and GLOW [6]; Flow++ [18]; nonlinear-squared flow [19]; linear, quadratic [20], cubic [21], and rational quadratic splines [22]; neural autoregressive flows [23], and unconstrained monotonic neural networks [24]. We confirm our theoretical findings experimentally and identify directions for further improvement.

2 Related work

Analyzing which distributions coupling-based normalizing flows can approximate is an active area of research. A general statement shows that a coupling-based normalizing flow which can approximate an arbitrary invertible function can learn any probability density *weakly* [1]. This applies to affine coupling flows [4, 5, 6], Flow++ [18], neural autoregressive flows [26], and SOS polynomial flows [27]. Affine coupling flows converge to arbitrary densities in Wasserstein distance [15]. Both universality results, however, require that the couplings become ill-conditioned (i.e. the learnt functions become increasingly discontinuous as the error decreases, whereas in practice one observes that functions remain smooth). Also, they consider only a finite subspace of the data space. Even more importantly, the convergence criterion employed in their proofs (weak convergence resp. convergence under Wasserstein metric) is critical: Those criteria do not imply convergence in the loss that is employed in practice [15, Remark 3], the Kullback-Leibler divergence (equivalent to maximum likelihood). An arbitrarily small distance in any of the above metrics can even result in an infinite KL divergence. In contrast to previous work on affine coupling flows, we work directly on the KL divergence. We decompose it in two contributions and show the flow's convergence for one of the parts.

Regarding when ill-conditioned flows need to arise to fit a distribution, [28] showed that well-conditioned affine couplings can approximate log-concave padded distributions, again in terms of Wasserstein distance. Lipschitz flows on the other hand cannot model arbitrary tail behavior, but this can be fixed by adapting the latent distribution [29].

SOS polynomial flows converge in *total variation* to arbitrary probability densities [30], which also does not imply convergence in KL divergence; zero-padded affine coupling flows converge weakly [23], and so do Neural ODEs [31, 32].

Closely related to our work, 48 linear affine coupling blocks can represent any invertible linear function $Ax + b$ with $\det(A) > 0$ [15, Theorem 2]. This also allows mapping any Gaussian distribution $\mathcal{N}(m, \Sigma)$ to the standard normal $\mathcal{N}(0, I)$. We put this statement into context in terms of the KL divergence: The loss is exactly composed of the divergence to the nearest Gaussian and of that Gaussian to the standard normal. We then make strong statements about the convergence of the latter, concluding that for typical flows a smaller number of layers is required for accurate approximation than predicted by [15].

3 Coupling-based normalizing flows

Normalizing flows learn an invertible function $f_\theta(x)$ that maps samples x from some unknown distribution $p(x)$ given by samples to *latent variables* $z = f_\theta(x)$ so that z follow a simple distribution, typically the standard normal. The function f_θ then yields an estimate $q(x)$ for the true data distribution $p(x)$ via the change of variables formula (e.g. [5]):

$$q(x) = \mathcal{N}(f_\theta(x); 0, I)|\det J|, \tag{1}$$

where $J = \nabla f_\theta(x)$ is the Jacobian of $f_\theta(x)$. We can train a normalizing flow via the maximum likelihood loss, which is equivalent to minimizing the Kullback-Leibler divergence between the distribution of the latent code $q(z)$, as given by $z = f_\theta(x)$ when $x \sim p(x)$, and the standard normal:

$$L = \mathcal{D}_{\mathrm{KL}}(q(z)\|\mathcal{N}(0, I)) = \mathbb{E}_{x\sim p(x)}\left[\tfrac{1}{2}\left\|f_\theta(x)\right\|^2 - \log|\det J|\right] + \mathrm{const}\,. \tag{2}$$

The invertible architecture that makes up f_θ has to (i) be computationally easy to invert, (ii) be able to represent complex transformations, and (iii) have a tractable Jacobian determinant $|\det J|$ [9]. Building such an architecture is an active area of research, see e.g. [2] for a review. In this work, we focus on the family of coupling-based normalizing flows, first presented in the form of the NICE architecture [4]. It is a deep architecture that consists of several blocks, each containing a rotation, a coupling and an ActNorm layer [6]:

$$f_{\text{block}}(x) = (f_{\text{act}} \circ f_{\text{cpl}} \circ f_{\text{rot}})(x). \tag{3}$$

The coupling f_{cpl} splits an incoming vector x_0 in two parts along the coordinate axis: The first part p_0, which we call *passive*, is left unchanged. The second part a_0, which we call *active*, is modified as

a function of the passive dimensions:

$$f_{\text{cpl}}(x_0) = f_{\text{cpl}}\begin{pmatrix} p_0 \\ a_0 \end{pmatrix} = \begin{pmatrix} p_0 \\ c(a_0; p_0) \end{pmatrix} =: \begin{pmatrix} p_1 \\ a_1 \end{pmatrix}. \tag{4}$$

Here, the coupling function $c : \mathbb{R}^{D/2} \times \mathbb{R}^{D/2} \to \mathbb{R}^{D/2}$ has to be a function that is easy to invert when p_0 is given, i.e. it is easy to compute $a_0 = c^{-1}(a_1; p_0)$ given p_0. This makes the coupling easy to invert: Call $x_1 = (p_1; a_1)$ the output of the layer, then $p_0 = p_1$. Use this to invert $a_1 = c(a_0; p_0)$. For example, RealNVP [5] proposes a simple affine transformation for c: $a_1 = c(a_0; p_0) = a_0 \odot s(p_0) + t(p_0)$ ($\odot$ means element-wise multiplication). $s(p_0) \in \mathbb{R}_+^{D/2}$ and $t(p_0) \in \mathbb{R}^{D/2}$ are computed by a feed-forward neural network. The coupling functions c of other architectures our theory applies to are listed in Appendix C.

An Activation Normalization (ActNorm) layer [6] helps stabilize training and is implemented in practice like in the popular INN framework `FrEIA` [33]. It rescales and shifts each dimension:

$$f_{\text{act}}(x) = r \odot x + u, \tag{5}$$

given parameters $r \in \mathbb{R}_+^D$ and $u \in \mathbb{R}^D$. We include it as it simplifies our mathematical arguments.

If we were to concatenate several coupling layers, the entire network would never change the passive dimensions apart from the element-wise affine transformation in the ActNorm layer. Here, the rotation layers $f_{\text{rot}}(x) = Qx$ come into play [6]. They multiply an orthogonal matrix Q to the data, changing which subspaces are passive respectively active. This matrix is typically fixed at random at initialization and then left unchanged during training.

4 Coupling layers as whitening transformation

The central mathematical question we answer in this work is the following: How can a deep coupling-based normalizing flow *whiten* the data? As the latent distribution is a standard normal, whitening is a necessary condition for the flow to converge. This is a direct property of the loss:

Proposition 1 (Pythagorean Identity, Proof in Appendix B.1)**.** *Given data with distribution $p(x)$ with mean m and covariance Σ. Then, the Kullback-Leibler divergence to a standard normal distribution decomposes as follows:*

$$\mathcal{D}_{KL}(p(x)\|\mathcal{N}(0, I)) = \underbrace{\mathcal{D}_{KL}(p(x)\|\mathcal{N}(m, \Sigma))}_{\textit{non-Gaussianity } \mathcal{G}(p)} + \underbrace{\mathcal{D}_{KL}(\mathcal{N}(m, \Sigma)\|\mathcal{N}(0, I))}_{\textit{non-Standardness } \mathcal{S}(p)} \tag{6}$$

and the non-Standardness again decomposes:

$$\mathcal{S}(p) = \underbrace{\mathcal{D}_{KL}(\mathcal{N}(m, \Sigma)\|\mathcal{N}(m, \operatorname{Diag}(\Sigma)))}_{\textit{Correlation } \mathcal{C}(p)} + \underbrace{\mathcal{D}_{KL}(\mathcal{N}(m, \operatorname{Diag}(\Sigma))\|\mathcal{N}(0, I))}_{\textit{Diagonal non-Standardness}}. \tag{7}$$

This splits the transport from the data distribution to the latent standard normal into three parts: (i) From the data to the nearest Gaussian distribution $\mathcal{N}(m, \Sigma)$, measured by $\mathcal{G}$. (ii) From that nearest Gaussian to the corresponding uncorrelated Gaussian $\mathcal{N}(m, \operatorname{Diag}(\Sigma))$, measured by $\mathcal{C}$. (iii) From the uncorrelated Gaussian to standard normal.

We do not make explicit use of the fact that the *non-Standardness* can again be decomposed, but we show it nevertheless to relate our result to the literature: The Pythagorean identity $\mathcal{D}_{\text{KL}}(p(x)\|\mathcal{N}(m, \operatorname{Diag}(\Sigma))) = \mathcal{G}(p) + \mathcal{C}(p)$ has been shown before by [25, Section 2.3]. Both their and our result are specific applications of the general [34, Theorem 3.8] from information geometry. Our proof is given in Appendix B.1.

Proposition 1 is visualized in Figure 1. In an experiment, we fit a set of Glow [6] coupling flows of increasing depths to the EMNIST digit dataset [35] using maximum likelihood loss and measure the capability of each flow in decreasing $\mathcal{G}$ and $\mathcal{S}$ (Details in Appendix A.1). The form of the non-Standardness $\mathcal{S}$ is given by the well-known KL divergence between the involved normal distributions, see Equation (30) in Appendix B.1. It is invariant under rotations Q and only depends on the first two moments m, Σ:

$$\mathcal{S}(m, \Sigma) := \mathcal{S}(p) = \frac{1}{2}(\|m\|^2 + \operatorname{tr}\Sigma - D - \log\det\Sigma)) = \mathcal{S}(Qm, Q\Sigma Q^{\mathrm{T}}). \tag{8}$$

The non-Standardness $\mathcal{S}$ will be our measure on how far the covariance and mean have approached the standard normal in the latent space. We give explicit loss guarantees for $\mathcal{S}$ for a single coupling block in Theorems 1 and 2 and imply a linear convergence rate for a deep network in Theorem 3.

Deep Normalizing Flows are typically trained end-to-end, i.e. the entire stack of blocks is trained jointly. In this work, our ansatz is to consider the effect of a single coupling block on the non-Standardness $\mathcal{S}$. Then, we combine the effect of many isolated blocks, disregarding potential further improvements to $\mathcal{S}$ due to joint, cooperative learning of all blocks. This simplifies the theoretical analysis of the network, but it is not a restriction on the model: Any function that is achieved in block-wise training could also be the solution of end-to-end training.

We aim to strongly reduce $\mathcal{S}$ while leaving room for a complementary theory explaining how non-Gaussianity $\mathcal{G}$ is reduced in practice. Note that affine-linear functions $Ax + b$ can never change $\mathcal{G}$, because they jointly transform the distribution $p(x)$ at hand and correspondingly the closest Gaussian to it (see Lemma 1 in Appendix B.2). Thus, if we restrict our coupling layers to be affine-linear functions, we are able to reduce $\mathcal{S}$ without increasing $\mathcal{G}$ in turn. This motivates considering affine-linear couplings of the following form, spelled out together with ActNorm as given by Equation (5). **The results in this work apply to all coupling architectures**, as they all can represent this coupling, see Appendix C.

$$\begin{pmatrix} p_1 \\ a_1 \end{pmatrix} = (f_{\text{act}} \circ f_{\text{cpl}})(Qx) = r \odot \begin{pmatrix} I & 0 \\ T & I \end{pmatrix} \begin{pmatrix} p_0 \\ a_0 \end{pmatrix} + u. \tag{9}$$

For future work considering $\mathcal{G}$, we propose to lift the restriction to affine-linear layers while making sure that $\mathcal{S}$ behaves as described in what follows. As the convergence of $\mathcal{G}$ however will strongly depend on the coupling architecture and data $p(x)$ at hand, this is beyond the scope of this work.

Our first result shows which mean m_1 and covariance Σ_1 a single affine-linear coupling as in Equation (9) yields to minimize $\mathcal{S}(m_1, \Sigma_1)$ given data with mean m and covariance Σ, rotated by Q:

Proposition 2 (Proof in Appendix B.2)**.** *Given D-dimensional data with mean m and covariance Σ and a rotation matrix Q. Split the covariance of the rotated data into four blocks, corresponding to the passive and active dimensions of the coupling layer:*

$$Q\Sigma Q^{\mathrm{T}} = \Sigma_0 = \begin{pmatrix} \Sigma_{0,pp} & \Sigma_{0,pa} \\ \Sigma_{0,ap} & \Sigma_{0,aa} \end{pmatrix} \tag{10}$$

Then, the moments m_1, Σ_1 that can be reached by a coupling as in Equation (9) *are:*

$$m_1 = 0, \qquad \Sigma_1 = \begin{pmatrix} M(\Sigma_{0,pp}) & 0 \\ 0 & M(\Sigma_{0,aa} - \Sigma_{0,ap}\Sigma_{0,pp}^{-1}\Sigma_{0,pa}) \end{pmatrix}. \tag{11}$$

This minimizes $\mathcal{S}$ as given in Equation (8)*, and $\mathcal{G}$ does not increase.*

The function M takes a matrix A and rescales the diagonal to 1 as follows. It is a well-known operation in numerics called Diagonal scaling or Jacobi preconditioning so that $M(A)_{ii} = 1$:

$$M(A)_{ij} = \sqrt{A_{ii}A_{jj}}^{-1} A_{ij} = (\mathrm{Diag}(A)^{-1/2} A \,\mathrm{Diag}(A)^{-1/2})_{ij}. \tag{12}$$

Proposition 2 shows how the covariance can be brought closer to the identity.

The new covariance has passive and active dimensions uncorrelated. In the active subspace, the covariance is the Schur complement $\Sigma_{0,aa} - \Sigma_{0,ap}\Sigma_{0,pp}^{-1}\Sigma_{0,pa}$. This coincides with the covariance of the Gaussian $\mathcal{N}(0, \Sigma)$ as it is conditioned on any passive value p. Afterwards, the diagonal is rescaled to one, matching the standard deviations of all dimensions with the desired latent code. The proof is based on a more general result how a single layer maximally reduces the Maximum Likelihood Loss for arbitrary data [14], which we apply to the non-Standardness $\mathcal{S}$ (see Appendix B.2).

Figure 2 shows an experiment in which a single affine-linear layer was trained to bring the covariance of EMNIST digits [35] as close to I as possible (Details in Appendix A.2). The experimental result coincides with the prediction by Proposition 2. Due to the finite batch-size, a small difference between theory and experiment remains.

5 Explicit convergence rate

In Section 4, we showed how a single coupling layer acts on the first two moments of a given data distribution to whiten it. We now explicitly demonstrate how much progress this means in terms

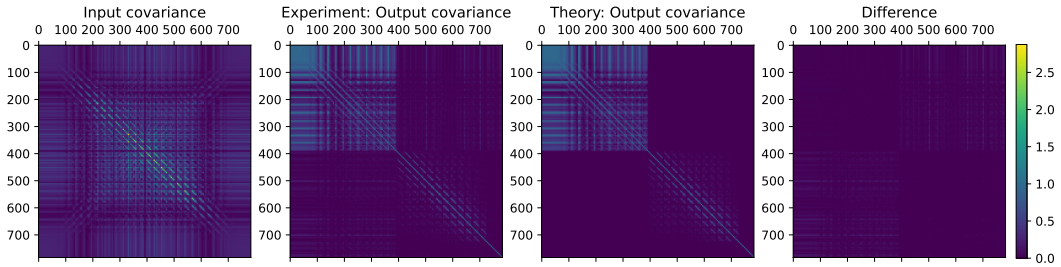

Figure 2: **How a single coupling layer can whiten the covariance** at the example of the EMNIST digits covariance matrix *(first panel)*. The covariance after a single layer trained experimentally to minimize non-Standardness $\mathcal{S}(m_1, \Sigma_1)$ *(second panel)*, which matches closely the prediction of Proposition 2 *(third panel)*. The difference between theory and experiment vanishes *(last panel)*.

of the non-Standardness $\mathcal{S}(m_1, \Sigma_1)$ averaged over rotations Q (Theorems 1 and 2) and show the consequences for multiple blocks (Theorem 3).

5.1 Single coupling block guarantees

Proposition 2 allows the computation of the minimum non-Standardness after a single coupling block given its rotation Q, by evaluating $\mathcal{S}(m_1, \Sigma_1)$. In fact, if we were to choose Q such that the data is rotated so that principal components lie on the axes (i.e. obtain Q using PCA), a single coupling block suffices to reduce the covariance to the identity: $\Sigma_0 = Q\Sigma Q^{\mathrm{T}}$ would be a diagonal matrix and $\Sigma_1 = I$. This is not the case in practice, where this optimal orientation has zero probability: Q is chosen uniformly at random before training from all orthogonal matrices. One could argue that one should whiten the data before passing it to the flow, reducing $\mathcal{S}$ to zero from the start. However, any change in the architecture could possibly alter the performance of the network with regard to reducing the non-Gaussianity $\mathcal{G}$. Also, our work shows that coupling-based normalizing flows are already well-equipped to bring the non-Standardness to zero without such modifications. To properly describe the achievable non-Standardness $\mathcal{S}$, **we formulate all guarantees as expectations over the rotation** Q, corresponding to the loss averaged over training runs.

We make two mild assumptions on our data that are part of usual data-preprocessing, when the mean is subtracted from the data and all data points are divided by the scalar $\sqrt{\operatorname{tr}\Sigma/D}$ (not to be confused with diagonal preconditioning, which acts dimension-wise).

Assumption 1. *The data $p(x)$ is centered:* $\mathbb{E}_{x\sim p(x)}[x] = 0$.

Assumption 2. *The covariance is normalized:* $\operatorname{tr}\Sigma = D$.

The assumptions simplify the non-Standardness in Equation (8), which now only depends on the determinant of Σ:

$$\mathcal{S}(\Sigma) = -\tfrac{1}{2}\log\det\Sigma = -\tfrac{1}{2}\log\det\Sigma_0 = \mathcal{S}(\Sigma_0) \tag{13}$$

for arbitrary rotation Q. We aim to compute the average non-Standardness after a single block $\mathbb{E}_{Q\in p(Q)}[\mathcal{S}(\Sigma_1(Q))]$. For any Q, $\mathcal{S}(\Sigma_1)$ is again given by the determinant of the covariance $\Sigma_1(Q)$ as Assumptions 1 and 2 remain fulfilled: By Proposition 2 $m_1 = 0$ and the diagonal preconditioning M ensures that the trace of Σ_1 is D. We write $\det(\Sigma_1)$ via M_a and M_p, the diagonal matrices that make up the diagonal preconditioning in Equation (12), and use the Schur determinantal formula for the determinant of block matrices: $\det(\Sigma_{0,pp})\det(\Sigma_{0,aa} - \Sigma_{0,ap}\Sigma_{0,pp}^{-1}\Sigma_{0,pa}) = \det(\Sigma_0) = \det(\Sigma)$ [36]. We thus get $\det(\Sigma_1) = \det(M_p\Sigma_{0,pp}M_p)\det(M_a(\Sigma_{0,aa} - \Sigma_{0,ap}\Sigma_{0,pp}^{-1}\Sigma_{0,pa})M_a) = \det(M_p^2)\det(M_a^2)\det(\Sigma)$. Inserting this into Equation (13), we find:

$$\mathcal{S}(\Sigma_1) = -\tfrac{1}{2}(\log\det\Sigma + \log\det M_p^2 + \log\det M_a^2) \leq \mathcal{S}(\Sigma_0) = \mathcal{S}(\Sigma). \tag{14}$$

The inequality $\mathcal{S}(\Sigma_1) \leq \mathcal{S}(\Sigma_0)$ holds because $\Sigma_1 = \Sigma_0$ is an admissible solution of the coupling layer optimization, but Σ_1 as given by Proposition 2 is a minimizer of $\mathcal{S}(\Sigma_1)$.

We average this quantity over training runs, i.e. over rotations Q:

$$\mathbb{E}_{Q\sim p(Q)}[\mathcal{S}(\Sigma_1)] = -\tfrac{1}{2}\big(\log\det\Sigma + \mathbb{E}_{Q\sim p(Q)}[\log\det M_p^2] + \mathbb{E}_{Q\sim p(Q)}[\log\det M_a^2]\big). \tag{15}$$

The main difficulty lies in the computation of $\mathbb{E}_{Q\sim p(Q)}[\log\det M_a^2]$. Here, we contribute the two strong statements Theorems 1 and 2 below.

5.1.1 Precise guarantee

The first result relies on projected orbital measures as developed by [37]. This theory describes the eigenvalues of submatrices of matrices in a random basis. We require such a result for integrating over $p(Q)$ in $\mathbb{E}_{Q\sim p(Q)}[\log\det M_a^2]$. In contrast to typical choices of $p(Q)$, the theory to this date only covers data rotated by unitary matrices.[1] To comply with [37], we make two more assumptions:

Assumption 3. *The distribution of rotations is the Haar measure over* unitary *matrices* $U(D)$.

Assumption 4. *The eigenvalues of the covariance matrix* Σ *are distinct:* $\lambda_i \neq \lambda_j$ *for* $i \neq j$.

One could think that the step from orthogonal to unitary rotations takes us far away from the scenario we want to consider. We will later observe empirically that the difference between averaging over unitary and orthogonal matrices is negligible. Technically, the covariance matrix remains positive definite, so the non-Standardness $\mathcal{S}$ is always real (see Appendix B.3.4). We will write $\mathbb{E}_{Q\sim U(D)}[\,\cdot\,]$ to denote expectations over unitary matrices.

Assumption 4 is typically satisfied when working with real data that are in 'general position'. We are now ready to **compute the average training performance** of a single coupling block:

Theorem 1 (Proof in Appendix B.3)**.** *Given* D*-dimensional data with covariance* Σ *with eigenvalues* $\lambda_1,\dots\lambda_D$*. Assume that Assumptions 1 to 4 hold. Then, after a single coupling block, the expected non-Standardness is bounded from above:*

$$\mathbb{E}_{Q\in U(D)}[\mathcal{S}(\Sigma_1(Q))] < \mathcal{S}(\Sigma) + \tfrac{D}{2}\log\left((-1)^{\frac{D}{2}+1}\sum_{i=1}^{D}\lambda_i^{1-\frac{D}{2}}\log(\lambda_i)R(\lambda_i^{-1};\lambda_{\neq i}^{-1})e_{\frac{D}{2}-1}(\lambda_{\neq i}^{-1})\right). \quad (16)$$

Here, $\lambda_{\neq i} := \{\lambda_1,\dots,\lambda_{i-1},\lambda_{i+1},\dots,\lambda_D\}$ *and* R, e_K *are given by:*

$$R(a;\{b_i\}_{i=1}^N) = \prod_{i=1}^{N}\frac{1}{a-b_i} \quad \textit{and} \quad e_K(\{b_i\}_{i=1}^N) = \sum_{0<i_1<\dots<i_K\leq N} b_{i_1}\cdots b_{i_K}. \quad (17)$$

Inequality (16) sharply bounds the expected non-Standardness that can be achieved by a single block. The only approximation made is an inequality which comes close to equality as the dimension D increases due to the concentration of the corresponding probability distribution.

Figure 3 shows an experiment confirming Theorem 1 (Details in Appendix A.3). We start with covariance matrices using parametrized eigenvalue spectra. On each, we first apply a single coupling block with random Q and train the coupling that maximally reduces $\mathcal{S}$ (Proposition 2). Then we iteratively append 32 additional blocks in the same manner, building a flow of that depth. We average the resulting empirical ratio $\mathcal{S}(\Sigma_1)/\mathcal{S}(\Sigma)$ over several *orthogonal* orientations Q of the rotation layer for each input covariance matrix. Then, we compare this to (i) experimentally averaging over *unitary* rotations and (ii) to the prediction by Theorem 1 and confirm that it is a valid and close upper bound. Details for replication and more examples can be found in Appendix A.3.

The proof explicitly integrates $\mathbb{E}[M_a^2]$ using [37] (see Appendix B.3). Numerically evaluating Equation (16) can be hard even for small D as the summands scale as $\mathcal{O}(\exp(D))$, but the overall sum scales as $\mathcal{O}(D)$. High values cancel due to R alternating in sign, and one requires arbitrary-precision floating point software to evaluate Equation (16).

5.1.2 Interpretable guarantee

The guarantee in Theorem 1 yields useful predictions, but it does not lend itself to further analysis: How does the bound behave over several coupling blocks? What is the behavior for varying dimension D? Also, Assumption 3 restricts formal reasoning as we are interested in averaging over orthogonal and not unitary rotations. Our second single-block guarantee depends only on simple metrics of the covariance. Moreover, we drop Assumptions 3 and 4, averaging over orthogonal, not unitary, Q:

[1] The only result known to us would yield predictions for $D = 2$ [38], whereas we are interested in large D.

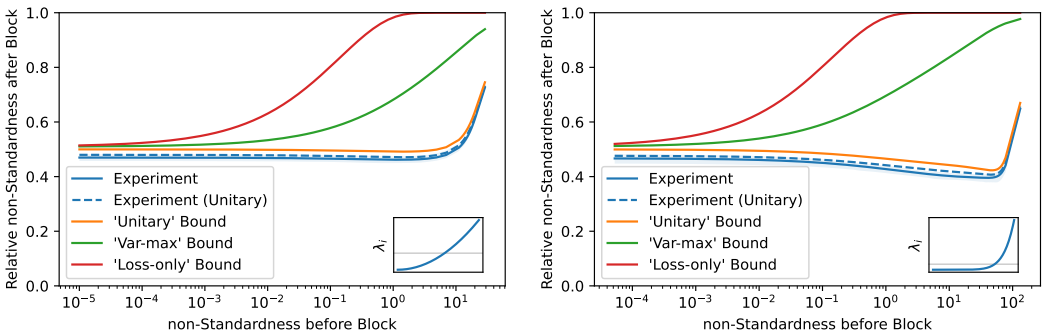

Figure 3: **Comparison between predicted non-Standardness and experiment** for 48-dimensional parametrized eigenvalue spectra *(insets)*, varied over a parameter which controls the spread of the spectrum and thus changes $\mathcal{S}$. The experimental average over *orthogonal* rotations matrices (blue, shaded by Interquartile Range IQR) is closely matched by the experimental average over *unitary* matrices (dotted blue). The prediction by Theorem 1 is a close upper bound that closely matches the experimental behavior (orange). The predictions by Theorem 2 are less precise, but converge to the same value as the precise bound for covariances close to the identitiy: 'Var-max' is Equation (18a) (green) and 'Loss-only' is Equation (18b) (red). More details and examples in Appendix A.3.

Theorem 2 (Proof in Appendix B.4)**.** *Given D-dimensional data fulfilling Assumptions 1 and 2 with covariance $\Sigma \neq I$ with eigenvalues $\lambda_1, \dots \lambda_D$. Then, after a single coupling block, the expected loss can be bounded from above:*

$$\mathbb{E}_{Q\in O(D)}[\mathcal{S}(\Sigma_1(Q))] \leq \mathcal{S}(\Sigma) + \frac{D}{4}\log\left(1 - \frac{D^2}{2(D-1)(D+2)}\frac{\mathrm{Var}[\lambda]}{\lambda_{\max}}\right) \tag{18a}$$

$$\leq \mathcal{S}(\Sigma) + \frac{D}{4}\log\left(1 - \frac{D^2}{(D-1)(D+2)}\frac{1-\sqrt{1-g^D}}{1+\sqrt{1-g^D}}(1-g)\right) < \mathcal{S}(\Sigma). \tag{18b}$$

Here, g is the geometric mean of the eigenvalues: $g = \prod_{i=1}^{D} \lambda_i^{1/D} = \exp(-2\mathcal{S}(\Sigma)/D) < 1$ which is a bijection of $\mathcal{S}(\Sigma)$.

These two new bounds on the average achievable non-Standardness $\mathcal{S}$ after a single block are also depicted in Figure 3. They make useful predictions, but are less precise than Theorem 1. The second bound will be especially useful in what follows because it only depends on the non-Standardness before the block $\mathcal{S}(\Sigma)$.

The full proof is given in Appendix B.4. It relies on the integration of monomials of entries of random orthogonal matrices as described by [39] and the arithmetic mean-geometric mean inequality by [40].

The first bound suggests an important property of the non-Standardness convergence of a coupling-based normalizing flow in terms of dimension: The performance only marginally depends on the dimension. To see this, divide Equation (18a) by D to obtain a statement about the non-Standardness per dimension $\mathcal{S}/D$. Then take several data sets with different dimension but same spectrum characteristics (i.e. same geometric mean, variance and maximum of covariance eigenvalues). The guarantee is then approximately constant in D (it varies slightly with $D^2/(D^2+D-2)$, which is always close to 1).

5.2 Deep network guarantee

The previous Section 5.1 was concerned with determining how much a *single* coupling block can typically contribute towards reducing the $\mathcal{S}$ to zero. Now, we extend this result to compute the expected non-Standardness after a *deep* coupling-based normalizing flow as an explicit function of the number of blocks. We again treat the rotation layer of each block as a random variable, as it is randomly determined before training.

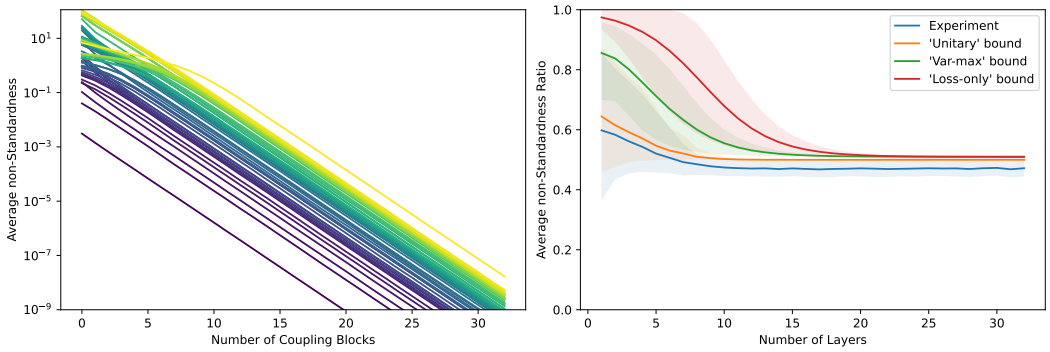

Figure 4: **Deep network convergence of covariance on toy dataset.** *(Left)* Each line shows the experimental convergence of $\mathcal{S}$ via the repeated application of Proposition 2, averaged over 32 runs with different rotations Q. *(Right)* The empirical convergence rate (blue), i.e. the ratio of $\mathcal{S}$ before and after a block, is correctly bounded from above by our predictions in Theorem 1 (orange), and the bounds in Theorem 2: Equation (18a) (green) and Equation (18b) (red). The solid lines show the ratio (bounds) averaged over the toy dataset and rotations, the shade is the IQR. The experiment suggests that a convergence rate like Theorem 3 can also be derived for the remaining bounds.

We find that the **convergence rate** of the covariance to the identity is (at least) **linear**:

Theorem 3 (Proof in Appendix B.5)**.** *Given D-dimensional data fulfilling Assumptions 1 and 2 with covariance Σ. Then, after L coupling blocks, the expected loss is smaller than:*

$$\mathbb{E}_{Q_1,\dots,Q_L \in O(D)}[\mathcal{S}(\Sigma_L)] \leq \gamma\big(\mathcal{S}(\Sigma)\big)^L \mathcal{S}(\Sigma), \tag{19}$$

where the convergence rate depends on the non-Standardness before training:

$$\gamma(\mathcal{S}) = 1 + \tfrac{1}{4\mathcal{S}/D} \log\left(1 - \frac{D^2}{(D-1)(D+2)} \frac{1-\sqrt{1-g(\mathcal{S})^D}}{1+\sqrt{1-g(\mathcal{S})^D}} \big(1 - g(\mathcal{S})\big)\right) < 1. \tag{20}$$

The non-Standardness decreases at least exponentially fast in the number of blocks. The convergence rate that holds for a deep network is computed using the non-Standardness of the input data $\mathcal{S}(\Sigma)$. This rate comes from Equation (18b). The proof uses that $\gamma(\mathcal{S})$ improves from block to block as $\mathcal{S}$ decreases (see Appendix B.5). Again, $g(\mathcal{S}) = \exp(-2\mathcal{S}/D) < 1$ is the geometric mean of eigenvalues of Σ, which increases from block to block.

Figure 4 shows the convergence of the non-Standardness to zero in an experiment. We build a toy dataset of various covariances where we aim to capture a plethora of possible cases (see Appendix A.4). We apply a single coupling block with random Q and the coupling that maximally reduces $\mathcal{S}$ via Proposition 2. We iteratively add such blocks 32 times, building a flow of that depth. The resulting convergence of $\mathcal{S}$ as a function of depth is averaged over 32 runs with different rotations. The measured curve confirms Theorem 3. We find that the rate γ in Equation (20) is correct, but several experiments show even faster convergence in practice. Indeed, the experiments suggest that dividing all upper bounds for $\mathbb{E}[\mathcal{S}(\Sigma_1)]$ in Theorems 1 and 2 by $\mathcal{S}(\Sigma)$ also bounds the non-Standardness ratio for subsequent blocks. Formally, we conjecture that $\mathbb{E}[\mathcal{S}(\Sigma_L)]/\mathcal{S}(\Sigma) \leq (B/\mathcal{S}(\Sigma))^L$ where B is the rhs. of Equations (16) and (18a) (Theorem 3 shows exactly this for Equation (18b)). We leave a proof or falsification of this conjecture open to future work.

The experiment also suggests that all bounds agree after a few blocks, leaving a small gap to the experiment. We can explicitly compute this limit value of $\gamma(\mathcal{S})$ by taking $\mathcal{S} \to 0$:

$$\gamma(\mathcal{S}) \xrightarrow{\mathcal{S}\to 0} \tfrac{D(D+2)-4}{2(D-1)(D+2)} \in \big[1/2, 5/9\big]. \tag{21}$$

The two experimental observations together with this limit value suggest the heuristic that **a single additional coupling block typically reduces the non-Standardness $\mathcal{S}$ by a factor of approximately 50%** if previous blocks are left unchanged, and possibly faster if cooperations between blocks are taken into account.

6 Conclusion

To the best of our knowledge, this is the first work on coupling-based normalizing flows that provides a quantitative convergence analysis in terms of the KL divergence. Specifically, a minimal convergence rate is established at which flows whiten the covariance of the input data under this strong measure of discrepancy of probability distributions. Splitting the loss into the non-Gaussianity $\mathcal{G}$ and the non-Standardness $\mathcal{S}$, we show that this whitening is a necessary condition for the flow to converge and give explicit guarantees. Our derivations suggest the rule of thumb that $\mathcal{S}$ can typically be reduced by about 50% per coupling block.

Our central idea was to separate out the contribution a single isolated block can make to reduce the loss, arguing that end-to-end training can only outperform the concatenation of isolated blocks.

Having separated the tasks a normalizing flow has to solve, and having explained how the non-Standardness $\mathcal{S}$ can be reduced to zero, we hope that explaining also the entire convergence of $\mathcal{L} = \mathcal{G} + \mathcal{S}$ with respect to the KL divergence is within reach. In particular, our theory did not yet explore how the non-linear part of each coupling block reduces the non-Gaussianity $\mathcal{G}$.

Acknowledgments and Disclosure of Funding

This work is supported by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC-2181/1 - 390900948 (the Heidelberg STRUCTURES Cluster of Excellence). It is also supported by the Vector Stiftung in the project TRINN (P2019-0092).

We thank our colleagues (in alphabetical order) Marcel Meyer, Jens Müller, Robert Schmier and Peter Sorrenson for their help and fruitful discussions.

References

[1] Takeshi Teshima, Isao Ishikawa, Koichi Tojo, Kenta Oono, Masahiro Ikeda, and Masashi Sugiyama. Coupling-based Invertible Neural Networks Are Universal Diffeomorphism Approximators. In *Advances in Neural Information Processing Systems*, 2020.

[2] Ivan Kobyzev, Simon J.D. Prince, and Marcus A. Brubaker. Normalizing Flows: An Introduction and Review of Current Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3964–3979, 2021.

[3] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing Flows for Probabilistic Modeling and Inference. *Journal of Machine Learning Research*, 57(22):1–64, 2021.

[4] Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: Non-linear Independent Components Estimation. In *International Conference on Learning Representations, Workshop Track*, 2015.

[5] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *International Conference on Learning Representations*, 2017.

[6] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, 2018.

[7] Radek Mackowiak, Lynton Ardizzone, Ullrich Kothe, and Carsten Rother. Generative classifiers as a basis for trustworthy image classification. In *IEEE / CVF Computer Vision and Pattern Recognition Conference*, 2021.

[8] Lynton Ardizzone, Radek Mackowiak, Carsten Rother, and Ullrich Köthe. Training Normalizing Flows with the Information Bottleneck for Competitive Generative Classification. In *Advances in Neural Information Processing Systems*, 2020.

[9] Lynton Ardizzone, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Analyzing Inverse Problems with Invertible Neural Networks. In *International Conference on Learning Representations*, 2018.

[10] Pablo Noever-Castelos, Lynton Ardizzone, and Claudio Balzani. Model updating of wind turbine blade cross sections with invertible neural networks. *Wind Energy*, 25(3):573–599, 2022.

[11] Jens Müller, Robert Schmier, Lynton Ardizzone, Carsten Rother, and Ullrich Köthe. Learning Robust Models Using The Principle of Independent Causal Mechanisms. In *German Conference on Pattern Recognition*, 2021.

[12] Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019.

[13] Tim J. Adler, Lynton Ardizzone, Anant Vemuri, Leonardo Ayala, Janek Gröhl, Thomas Kirchner, Sebastian Wirkert, Jakob Kruse, Carsten Rother, Ullrich Köthe, and Lena Maier-Hein. Uncertainty-aware performance assessment of optical imaging modalities with invertible neural networks. *International Journal of Computer Assisted Radiology and Surgery*, 14(6):997–1007, 2019.

[14] Felix Draxler, Jonathan Schwarz, Christoph Schnörr, and Ullrich Köthe. Characterizing the Role of a Single Coupling Layer in Affine Normalizing Flows. In *German Conference on Pattern Recognition*, 2020.

[15] Frederic Koehler, Viraj Mehta, and Andrej Risteski. Representational aspects of depth and conditioning in normalizing flows. In *International Conference on Machine Learning*, 2021.

[16] Alison L. Gibbs and Francis Edward Su. On Choosing and Bounding Probability Metrics. *International Statistical Review / Revue Internationale de Statistique*, 70(3):419–435, 2002.

[17] P. J. Huber and E. M. Ronchetti. *Robust Statistics*. John Wiley & Sons, Inc., second edition, 2009.

[18] Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design. In *International Conference on Machine Learning*, 2019.

[19] Zachary Ziegler and Alexander Rush. Latent Normalizing Flows for Discrete Sequences. In *International Conference on Machine Learning*, 2019.

[20] Thomas Müller, Brian Mcwilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. Neural Importance Sampling. *ACM Transactions on Graphics*, 38(5):1–19, 2019.

[21] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Cubic-Spline Flows. In *International Conference on Machine Learning, Workshop Track*, 2019.

[22] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural Spline Flows. In *Advances in Neural Information Processing Systems*, 2019.

[23] Chin-Wei Huang, Laurent Dinh, and Aaron Courville. Augmented Normalizing Flows: Bridging the Gap Between Generative Flows and Latent Variable Models. In *International Conference on Learning Representations, Workshop Track*, 2020.

[24] Antoine Wehenkel and Gilles Louppe. Unconstrained Monotonic Neural Networks. In *Advances in Neural Information Processing Systems*, 2019.

[25] Jean-François Cardoso. Dependence, Correlation and Gaussianity in Independent Component Analysis. *Journal of Machine Learning Research*, 4:1177–1203, 2003.

[26] Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural Autoregressive Flows. In *International Conference on Machine Learning*, 2018.

[27] Priyank Jaini, Kira A. Selby, and Yaoliang Yu. Sum-of-Squares Polynomial Flow. In *International Conference on Machine Learning*, 2019.

[28] Holden Lee, Chirag Pabbaraju, Anish Sevekari, and Andrej Risteski. Universal Approximation for Log-concave Distributions using Well-conditioned Normalizing Flows. In *International Conference on Machine Learning, Workshop Track*, 2021.

[29] Priyank Jaini, Ivan Kobyzev, Yaoliang Yu, and Marcus Brubaker. Tails of Lipschitz Triangular Flows. In *International Conference on Machine Learning*, 2020.

[30] Isao Ishikawa, Takeshi Teshima, Koichi Tojo, Kenta Oono, Masahiro Ikeda, and Masashi Sugiyama. Universal approximation property of invertible neural networks. *arXiv preprint arXiv:2204.07415*, 2022.

[31] Han Zhang, Xi Gao, Jacob Unterman, and Tom Arodz. Approximation Capabilities of Neural ODEs and Invertible Residual Networks. In *International Conference on Machine Learning*, 2020.

[32] Takeshi Teshima, Koichi Tojo, Masahiro Ikeda, Isao Ishikawa, and Kenta Oono. Universal Approximation Property of Neural Ordinary Differential Equations. In *Advances in Neural Information Processing Systems, Workshop Track*, 2020.

[33] Lynton Ardizzone, Till Bungert, Felix Draxler, Ullrich Köthe, Jakob Kruse, Robert Schmier, and Peter Sorrenson. Framework for Easily Invertible Architectures (FrEIA). https://github.com/VLL-HD/FrEIA, 2018.

[34] Shun-ichi Amari and Hiroshi Nagaoka. *Methods of Information Geometry*. American Mathematical Society, 2007.

[35] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EMNIST: An extension of MNIST to handwritten letters. *arXiv preprint arXiv:1702.05373*, 2017.

[36] Roger A Horn and Charles R Johnson. *Matrix Analysis*. Cambridge University Press, 2012.

[37] Grigori Olshanski. Projections of orbital measures, Gelfand-Tsetlin polytopes, and splines. *Journal of Lie Theory*, 23(4):1011–1022, 2013.

[38] Jacques Faraut. Rayleigh theorem, projection of orbital measures and spline functions. *Advances in Pure and Applied Mathematics*, 6(4):261–283, 2015.

[39] T. Gorin. Integrals of monomials over the orthogonal group. *Journal of Mathematical Physics*, 43(6):3342–3351, 2002.

[40] D. I. Cartwright and M. J. Field. A refinement of the arithmetic mean-geometric mean inequality. *Proceedings of the American Mathematical Society*, 71(1):36–38, 1978.

[41] P.R. Krishnaiah. Some recent developments on complex multivariate distributions. *Journal of Multivariate Analysis*, 6(1):1–30, 1976.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] see Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Generative modeling, which this paper aims to improve, can be used in harmful ways to generate Deepfakes for disinformation.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] Full proofs are in Appendix B.

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Appendix A.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix A.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix A.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Whitening Convergence Rate of Coupling-based Normalizing Flows: Appendix

A Details on Experiments

All experiments were carried out on a single AMD Ryzen 7 3700X 8-Core Processor together with a NVIDIA GeForce RTX 2080. At `https://github.com/VLL-HD/Coupling-Flow-Bound` we have made available the code for all experiments.

A.1 Deep network on EMNIST

In this experiment, we estimate the capability of affine normalizing flows in reducing the non-Standardness $\mathcal{S}$ (see Equation (8)) as a function of the number of layers. We compare this to the theoretic bound in Theorem 1.

To this end, we train affine normalizing flows on EMNIST digits [35]. We leverage a 20-block Glow architecture as described in Section 3. To measure the effect of depth $L = 1, \ldots, 20$ of the flow on $\mathcal{S}$, we truncate the architecture to L layers.

The architecture is built as follows: We start by down-sampling the input image from gray scale $1 \times 28 \times 28$ to $4 \times 14 \times 14$: Each group of four neighboring pixels is reordered into one pixel with four times the channels in a checkerboard-like pattern. Then, eight convolutional coupling blocks with 16 hidden channels are applied. They are followed by another down-sampling to $16 \times 7 \times 7$ and eight convolutional coupling blocks with 32 hidden channels. After flattening the input, four fully-connected affine coupling blocks are added with 392 hidden dimensions.

When truncating this architecture, we remove blocks *from the left*. For example, when one block is present ($L = 1$), only the last coupling block with the fully connected subnetwork remains. This makes the theory in this paper applicable, as Proposition 2 assumes that the neural networks s and t are fully connected (otherwise, the whitening operation cannot always be represented).

We train each depth from scratch for 300 epochs using Adam with a learning rate of $3 \cdot 10^{-3}$ which is reduced by a factor of .1 after 100 and 200 epochs. The batch size is 240 which implies 1000 iterations per epoch.

Given the 20 networks of different, we split the loss into the non-Gaussianity $\mathcal{G}$ and non-Standardness $\mathcal{S}$ as suggested by Proposition 1. To do so, we compute the empirical covariances Σ_l of 10'000 test samples pushed through each flow.

To relate this experiment to our theory, we take the covariance matrices obtained using the trained flows Σ_l and apply Theorem 1 on each. This yields an upper bound on the expected non-Standardness after training another network with depth increased by one. In other words, given Σ_l, Theorem 1 predicts an upper bound on the expected $\mathbb{E}_{Q_{l+1} \sim p(Q)}[\mathcal{S}(\Sigma_{l+1})]$. We observe that the experimentally observed non-Standardness behaves similar to the upper bound. We do not expect this to be the case in general: There might be a trade-off between reducing $\mathcal{S}$ and $\mathcal{G}$, so the optimization might actually decide for reducing $\mathcal{G}$ at the cost of increasing $\mathcal{S}$. We only show that with the covariance in Proposition 2, $\mathcal{G}$ does not increase. On the other end, an affine flow might actually be able to reduce the non-Standardness stronger than predicted, as our theory does not take potentially useful cooperation between layers into account.

We average all results over eight runs per depth (i.e. $8 \cdot 20 = 160$ networks in total). Despite different random orientations in each run, the results are very concentrated: We find error bars so small that they are not visible in Figure 1.

We observe that after three blocks, the non-Standardness is close to zero. Here, the flow consists of one convolutional and two fully connected coupling blocks. This justifies the use of convolutional networks for s and t; it could have happened that the convolutional layer, being limited to reduce correlations between pixels only locally, does not reduce the non-Standardness as strongly as predicted. This justifies the use of convolutional layers for the remaining blocks.

Figure 5 shows samples from one networks trained for each depth (sampling temperature $T = 0.7$).

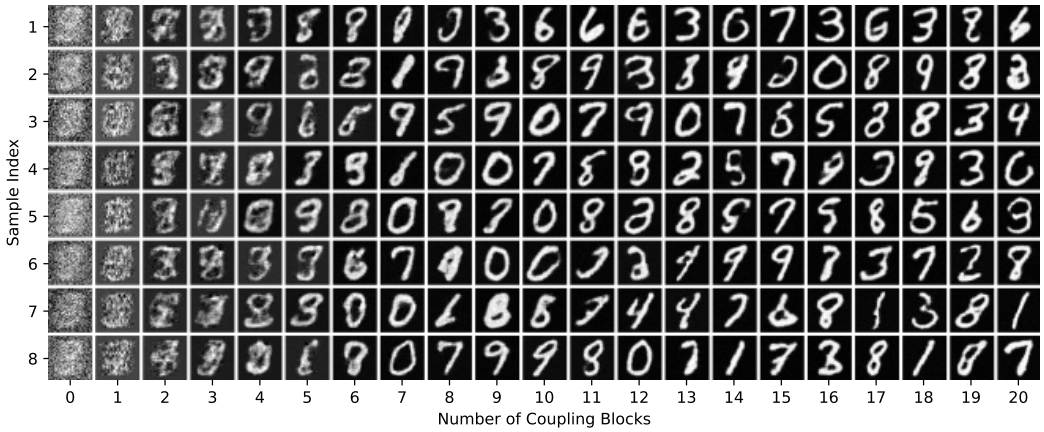

Figure 5: Samples generated by the affine coupling flows with varying depth trained for Figure 1. Each column shows eight samples by a network of the corresponding depth.

A.2 Single layer on EMNIST digit covariance

This experiment confirms that the covariance minimizing the non-Standardness $\mathcal{S}(\Sigma_1)$ after a single layer is correctly predicted by Proposition 2.

To get an interesting covariance matrix, we flatten the EMNIST digits training data and compute its covariance matrix Σ, as depicted in Figure 2 on the left. We then sample a multivariate Gaussian with this covariance matrix and train a single affine coupling layer. As the data is Gaussian, we can train with the standard maximum likelihood loss as it is equivalent to the non-Standardness $\mathcal{S}$. We use Adam with a learning rate 0.05, a batch size of 2048 and train for 512 iterations.

A.3 Single block on toy data

This experiment explores the average non-Standardness that can be reached by a single layer by modifying the covariance as given by Proposition 2. It also aims to confirm the upper bounds shown in Theorems 1 and 2.

We build a family of toy covariance matrices to work with. As the data will be randomly rotated anyway, we choose the matrices to be diagonal w.l.o.g., i.e. we directly design the eigenvalue spectrum of each covariance. We prescribe this spectrum by a continuous function $\mu : [0, 1] \to \mathbb{R}_+$. It is chosen bijective to ensure that the eigenvalues are distinct. We then define the eigenvalues as follows:

$$\mu_i = \mu\big(\tfrac{i}{D-1}\big) \quad i = 0, \dots, D-1. \tag{22}$$

With this approach, we can systematically modify eigenvalue/noise spectra.

Given a vector of eigenvalues $(\mu_i)_i$, we need to ensure that its mean is one. We do so by dividing by the mean:

$$\nu_i := \frac{\mu_i}{\sum_{i=1}^{D} \mu_i / D}. \tag{23}$$

Finally, we add a scaling parameter $s > 0$ that defines how close the spectrum is to the identity:

$$\lambda_i^{(s)} := (\nu_i - 1) \cdot s + 1. \tag{24}$$

The non-Standardness strictly decreases as s comes closer to 0. As the eigenvalues always have to be positive, s must be chosen smaller than:

$$s < \frac{1}{1 - \lambda_{\min}} =: s_{\max}. \tag{25}$$

Given a spectrum $\lambda_i^{(s)}$, we build a diagonal covariance matrix

$$\Sigma = \mathrm{Diag}(\lambda_i^{(s)})_{i=1}^{D}. \tag{26}$$

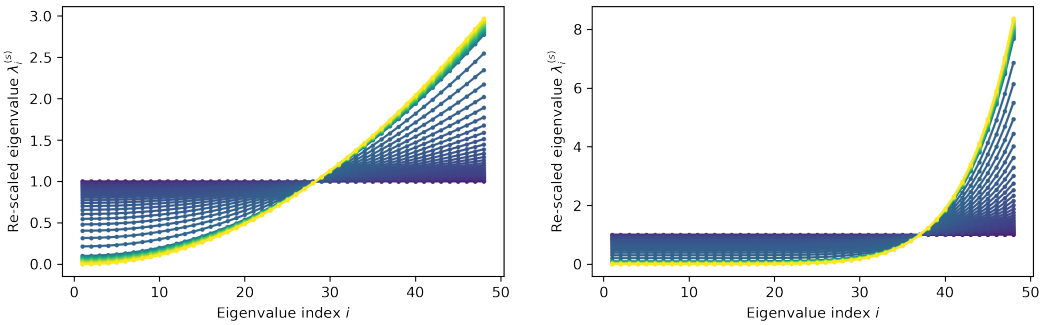

Figure 6: Eigenvalue spectra used for experiment depicted in Figure 3. *(Left)* $\mu(x) = x^2$ and *(right)* $\mu(x) = x^8$. Each line corresponds to a different scaling s.

For the experimental baseline, we sample N_{rot} orthogonal and unitary rotation matrices $Q \sim p(Q)$ from the corresponding Haar measure over $O(D)$ and $U(D)$. We employ `scipy.stats.ortho_group` respectively `scipy.stats.unitary_group`. This yields the covariance of the rotated data:

$$\Sigma_0 = Q\Sigma Q^T. \tag{27}$$

(Or, Q^* instead of Q^T if we average over unitary matrices).

We do not train affine coupling layers directly. Instead, we make use of the single layer output covariance Σ_1 from Proposition 2.

We choose the following numerical values for s: To get a close look at the case where $s \to 0$ and correspondingly $\mathcal{S} \to 0$, we take $N_{\text{scale}}/3$ geometrically spaced points in $[0.001s_{\max}, 0.9s_{\max}]$. To accurately capture the off-minimum behavior, we add to that $2N_{\text{scale}}/3$ linearly spaced points between $[0.9s_{\max}, .999s_{\max}]$.

We choose $N_{\text{rot}} = 100$ and $N_{\text{scale}} = 150$ for all experiments. To save computational resources, we re-use the rotations sampled for the first scale for the remaining.

In Figure 3, we showed the experiment for the parameterized spectra $\mu(x) = x^2$ and $\mu(x) = x^8$. For both, Figure 6 shows which rescaled eigenvalue spectra were used in this experiment. In Figure 7, we give examples for more spectra.

A.4 Layer-wise training on toy data

In this experiment, we track the non-Standardness as layers are added, check Theorem 3, and compare the convergence rate Equation (20) to the other bounds in Theorems 1 and 2.

This experiment uses a different set of toy covariances than Appendix A.3. This time, we build a plethora of different initial covariances (eigenvalue spectra) that include extreme cases:

1. All eigenvalues are set to 1 except for one that is varying.
2. All eigenvalues have the same value that is varied, except for one that is set to 1.
3. Split the eigenvalues into two halves, respectively having the same value: The first half is varied, the second half assume the inverse value of the first half.
4. Randomly sample all eigenvalues uniformly from $[0, 2]$.
5. Randomly sample all eigenvalues between such that the logarithm is uniformly distributed over $[1/v_{\max}, v_{\max}]$.

Whenever we vary the value of any eigenvalue, we take N_{vary} scalars geometrically spaced between $1/v_{\max}$ and $v_{\max}$. We exclude the case where all eigenvalues are equal to 1, implying a non-Standardness of 0.

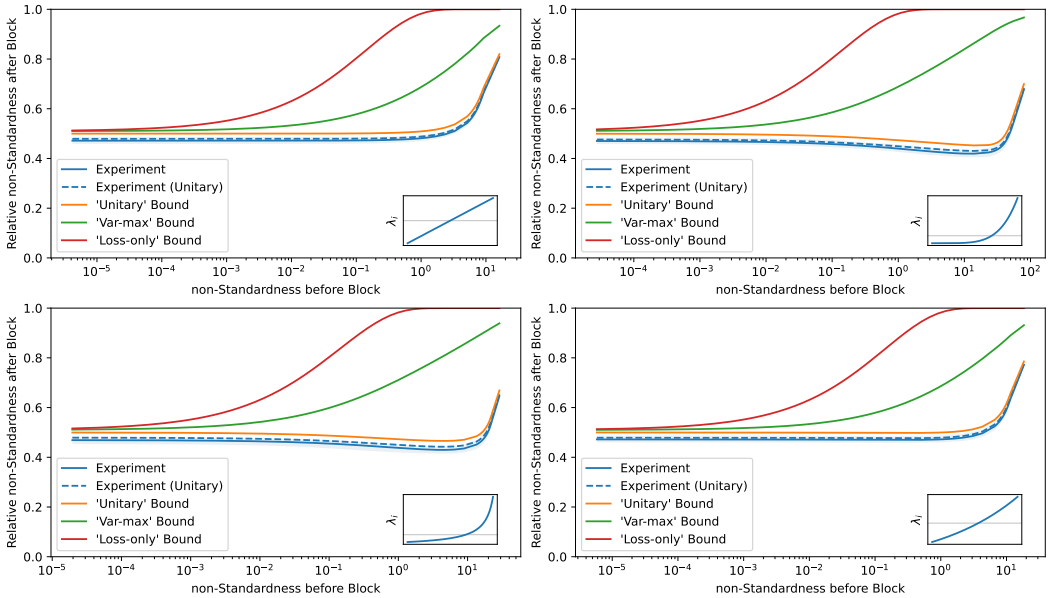

Figure 7: Examples for single layer relative non-Standardness on more eigenvalue spectra: *(Top left)* $\mu(x) = x$, *(top right)* $\mu(x) = x^5$, *(bottom left)* $\mu(x) = \frac{1}{1.1-x}$, *(bottom right)* $\mu(x) = \exp(x)$. More details in Figure 3.

To fulfill Assumption 4, we do not actually assign the same value to eigenvalues, but multiply them each with a linearly increasing factor in $(1-\epsilon, 1+\epsilon)$. We do not observe any change in experimental behavior from this, but this allows us evaluating Theorem 1.

Given the dataset of eigenvalues, we build diagonal covariances, repeatedly apply random rotations and the whitening procedure in Proposition 2. The details are given in Algorithm 1. For each input covariance, we obtain N_{rot} trajectories of covariances.

Algorithm 1 Multi-layer non-Standardness experiment

Require: Input covariances $\Sigma^{(i)}, i = 1, \dots, N$, number of rotations N_{rot}, number of layers L.
  $\Sigma_0^{(i,r)} \leftarrow \Sigma^{(i)}$ **for** $i = 1, \dots, N; r = 1, \dots, N_{\text{rot}}$ {Copy each input covariance N_{rot} times}
  for $l = 1, \dots, L$ **do**
    $Q^{(r)} \sim O(D)$ **for** $r = 1, \dots, N_{\text{rot}}$ {Sample rotations}
    $\Sigma_{l-1}^{(i,r)\prime} \leftarrow Q^{(r)} \Sigma_{l-1}^{(i,r)} (Q^{(r)})^{\mathrm{T}}$ **for** $i = 1, \dots, N; r = 1, \dots, N_{\text{rot}}$ {Apply rotations}
    $\Sigma_l^{(i,r)} \leftarrow$ Proposition 2 on $\Sigma_{l-1}^{(i,r)\prime}$ {Apply whitening step}
  end for
Ensure: $\{\Sigma_l^{(i,r)}\}_{l=1}^{L}$ **for** $i = 1, \dots, N; r = 1, \dots, N_{\text{rot}}$.

We evaluate the non-Standardness of each covariance matrix $\mathcal{S}(\Sigma_l^{(i,r)})$ and average over rotations. This is shown in the left plot in Figure 4.

In addition, we compute the relative non-Standardness between layers:

$$\mathcal{S}(\Sigma_l^{(i,r)}) / \mathcal{S}(\Sigma_{l-1}^{(i,r)}). \tag{28}$$

This quantity is averaged over rotations r and instances i. It is depicted together with the corresponding interquartile range (IQR) in the right half of Figure 4.

We also evaluate each of the bounds on $\mathbb{E}_Q[\mathcal{S}(\Sigma_{l+1}^{(i,r)})]$ in Theorems 1 and 2 given $\Sigma_l^{(i,r)}$ and divide it by the non-Standardness $\mathcal{S}(\Sigma_l^{(i,r)})$. Again, we average over rotations and iterations.

Averaging over rotations might be counter-intuitive as the bounds explicitly calculate a value that is an average: It is necessary because for each initial covariance, we have N_{rot} trajectories with different convergence behavior. Let us make this explicit. Denote by B any of the bounds in Theorems 1 and 2:

$$\frac{\mathbb{E}_{Q_{l+1}\sim p(Q)}[\mathcal{S}(\Sigma_{l+1}^{(i,r)}(Q_{l+1}))]}{\mathcal{S}(\Sigma_l^{(i,r)})} \leq \frac{B(\Sigma_l^{(i,r)})}{\mathcal{S}(\Sigma_l^{(i,r)})}. \tag{29}$$

We average the quantity on the right over the different trajectories, i.e. over i, r. It only depends on the covariances in the lth layer in contrast to the expression on the left.

As hyperparameters to the experiment, we choose $D = 48, L = 32, N_{\text{vary}} = 128, N_{\text{rot}} = 32, v_{\max} = 1000, \epsilon = 10^{-5}$. We stop a trajectory once the non-Standardness falls below 10^{-9} to avoid numerical instabilities.

B Detailed proofs

B.1 Proof of Proposition 1

The explicit form of the non-Standardness is given by the KL divergence between the two multivariate Gaussians $\mathcal{N}(m, \Sigma)$ and $\mathcal{N}(0, I)$:

$$\begin{aligned}
&\mathcal{D}_{\text{KL}}(\mathcal{N}(m,\Sigma)\|\mathcal{N}(0,I)) && \text{(30a)}\\
&= \mathbb{E}_{x\sim\mathcal{N}(m,\Sigma)}[\log\mathcal{N}(x;m,\Sigma) - \log\mathcal{N}(x;0,I)] && \text{(30b)}\\
&= \mathbb{E}_{x\sim\mathcal{N}(m,\Sigma)}[-\tfrac{1}{2}\log\det(2\pi\Sigma) - \tfrac{1}{2}(x-m)^{\mathrm{T}}\Sigma^{-1}(x-m) + \tfrac{1}{2}\log\det(2\pi I_D) + \tfrac{1}{2}\|x\|^2] && \text{(30c)}\\
&= \tfrac{1}{2}\Big(-\log\det(\Sigma) + \mathbb{E}_{x\sim\mathcal{N}(m,\Sigma)}[-\tfrac{1}{2}(x-m)^{\mathrm{T}}\Sigma^{-1}(x-m) + \tfrac{1}{2}\|x\|^2]\Big) && \text{(30d)}\\
&= \tfrac{1}{2}(\|m\|^2 + \operatorname{tr}\Sigma - D - \log\det\Sigma). && \text{(30e)}
\end{aligned}$$

Proof. We start with the first decomposition in Equation (6).

$$\begin{aligned}
&\mathcal{D}_{\text{KL}}(p\|\mathcal{N}(0,I)) - \mathcal{D}_{\text{KL}}(p\|\mathcal{N}(m,\Sigma)) && \text{(31a)}\\
&= \mathbb{E}_{x\sim p(x)}[\log p(x) - \log\mathcal{N}(x;0,I) - \log p(x) + \log\mathcal{N}(x;m,\Sigma)] && \text{(31b)}\\
&= \mathbb{E}_{x\sim p(x)}[-\log\mathcal{N}(x;0,I) + \log\mathcal{N}(x;m,\Sigma)] && \text{(31c)}\\
&= \frac{1}{2}\mathbb{E}_{x\sim p(x)}[D\log(2\pi) + \|x\|^2 - D\log(2\pi) - \log\det\Sigma - (x-m)^{\mathrm{T}}\Sigma^{-1}(x-m)] && \text{(31d)}\\
&= \frac{1}{2}\mathbb{E}_{x\sim p(x)}[\|x\|^2 - \log\det\Sigma - (x-m)^{\mathrm{T}}\Sigma^{-1}(x-m)] && \text{(31e)}\\
&= \frac{1}{2}\Big(\mathbb{E}_{x\sim p(x)}[\|x\|^2] - \log\det\Sigma - \mathbb{E}_{x\sim p(x)}[(x-m)^{\mathrm{T}}\Sigma^{-1}(x-m)]\Big). && \text{(31f)}
\end{aligned}$$

The open expectation values read:

$$\mathbb{E}_{x\sim p(x)}[\|x\|^2] = \mathbb{E}_{x\sim p(x)}[\sum_{i=1}^{D} x_i^2] = \sum_{i=1}^{D}\mathbb{E}_{x\sim p(x)}[x_i^2] = \sum_{i=1}^{D}(m_i^2 + \Sigma_{ii}) = \|m\|^2 + \operatorname{tr}\Sigma, \tag{32}$$

and interpreting $(x - m)$ as a $\mathbb{R}^{D\times 1}$ matrix, we can re-write using the trace. Then use the cyclic property and linearity of the trace:

$$\begin{aligned}
\mathbb{E}_{x\sim p(x)}[(x-m)^{\mathrm{T}}\Sigma^{-1}(x-m)] &= \mathbb{E}_{x\sim p(x)}[\operatorname{tr}((x-m)^{\mathrm{T}}\Sigma^{-1}(x-m))] && \text{(33a)}\\
&= \mathbb{E}_{x\sim p(x)}[\operatorname{tr}((x-m)(x-m)^{\mathrm{T}}\Sigma^{-1})] && \text{(33b)}\\
&= \operatorname{tr}(\mathbb{E}_{x\sim p(x)}[(x-m)(x-m)^{\mathrm{T}}]\Sigma^{-1}) && \text{(33c)}\\
&= \operatorname{tr}(\Sigma\Sigma^{-1}) && \text{(33d)}\\
&= D. && \text{(33e)}
\end{aligned}$$

Inserting the two expectation values, we identify:

$$\mathcal{D}_{\mathrm{KL}}(p\|\mathcal{N}(0,I)) - \mathcal{D}_{\mathrm{KL}}(p\|\mathcal{N}(m,\Sigma)) = \frac{1}{2}\Big(\|m\|^2 + \operatorname{tr}\Sigma - \log\det\Sigma - D\Big) \tag{34a}$$

$$= \mathcal{D}_{\mathrm{KL}}(\mathcal{N}(m,\Sigma)\|\mathcal{N}(0,I)), \tag{34b}$$

and obtain Equation (6).

Now we move on to show Equation (7):

$$\mathcal{C}(p) = \mathcal{D}_{\mathrm{KL}}(\mathcal{N}(m,\Sigma)\|\mathcal{N}(m,\operatorname{Diag}(\Sigma))) \tag{35a}$$

$$= \frac{1}{2}\left(\operatorname{tr}\Big((\operatorname{Diag}\Sigma)^{-1}\Sigma\Big) - D + \log\frac{\det(\operatorname{Diag}(\Sigma))}{\det\Sigma}\right) \tag{35b}$$

$$= \frac{1}{2}\log\frac{\det(\operatorname{Diag}(\Sigma))}{\det\Sigma}, \tag{35c}$$

and

$$\mathcal{D}_{\mathrm{KL}}(\mathcal{N}(m,\operatorname{Diag}(\Sigma))\|\mathcal{N}(0,I)) = \frac{1}{2}\big(\operatorname{tr}\operatorname{Diag}(\Sigma) - D - \log(\det(\operatorname{Diag}(\Sigma)))\big) \tag{36a}$$

$$= \frac{1}{2}\big(\operatorname{tr}\Sigma - D - \log(\det(\operatorname{Diag}(\Sigma)))\big). \tag{36b}$$

Adding the two divergences yields Equation (7). □

B.2 Proof of Proposition 2

We first show that an affine-linear function $g(x)$ as assumed in Proposition 2 cannot change the non-Gaussianity $\mathcal{G}$:

Lemma 1. *Given a D-dimensional distribution and an affine-linear function*

$$g(x) = Ax + b \tag{37}$$

for some $A \in \mathbb{R}^{D\times D}$ with $\det A > 0$ and $b \in \mathbb{R}^D$. Then:

$$\mathcal{G}(g_\sharp p) = \mathcal{G}(p). \tag{38}$$

Proof. The non-Gaussianity $\mathcal{G}$ is given by:

$$\mathcal{G}(p) = \mathcal{D}_{\mathrm{KL}}(p(x)\|\mathcal{N}(m,\Sigma)). \tag{39}$$

Mean and covariance of the push-forward of p via g read:

$$\mathbb{E}_{x\sim p(x)}[g(x)] = \mathbb{E}_{x\sim p(x)}[Ax+b] = Am + b = m_1, \tag{40a}$$

$$\operatorname{Cov}_{x\sim p(x)}[g(x)] = \operatorname{Cov}_{x\sim p(x)}[Ax+b] = A\Sigma A^{\mathrm{T}} = \Sigma_1. \tag{40b}$$

Thus, the non-Gaussianity after applying g reads:

$$\mathcal{G}(g_\sharp p) = \mathcal{D}_{\mathrm{KL}}(g_\sharp p\|\mathcal{N}(m_1,\Sigma_1)). \tag{41}$$

The push-forward of $\mathcal{N}(m,\Sigma)$ via g is identical to the normal distribution that occurs in the non-Gaussianity of $g_\sharp p$:

$$g_\sharp\mathcal{N}(m,\Sigma) = \mathcal{N}(m_1,\Sigma_1), \tag{42}$$

Now, we make use of the fact that the KL divergence is invariant if both arguments are transformed by any invertible function g:

$$\mathcal{D}_{\mathrm{KL}}(p_1(x)\|p_2(x)) = \mathcal{D}_{\mathrm{KL}}((g_\sharp p_1)(x)\|(g_\sharp p_2)(x)). \tag{43}$$

Together,

$$\mathcal{G}(g_\sharp p) = \mathcal{G}(p). \tag{44}$$

□

We now turn to the **proof of Proposition 2**:

Proof. We aim to find the affine-linear coupling layer f_{cpl} minimizing $\mathcal{S}(\Sigma_1)$. By Lemma 1, $\mathcal{G}$ does not change.

The affine-linear coupling f_{cpl} has the following form:

$$x_1 = \begin{pmatrix} \text{Diag}(r) & 0 \\ T & \text{Diag}(s) \end{pmatrix} \begin{pmatrix} p_0 \\ a_0 \end{pmatrix} + \begin{pmatrix} u \\ t \end{pmatrix} =: Ax_0 + b. \tag{45}$$

To make the coupling affine-linear, $r, s \in \mathbb{R}_+^{D/2}$ are positive vectors, $u, t \in \mathbb{R}^{D/2}$ are vectors and $T \in \mathbb{R}^{D/2 \times D/2}$ is the matrix describing the linear dependence of a_1 on p_0.

By linearity of expectation, the mean of x_1 reads:

$$m_1 = Am_0 + b. \tag{46}$$

Write $S := \text{Diag}(s)$ and $R := \text{Diag}(r)$ so that the covariance of x_1 is given by:

$$\begin{aligned} \Sigma_1 := \text{Cov}[x_1] &= A\Sigma_0 A^{\text{T}} && \text{(47a)} \\ &= \begin{pmatrix} R & 0 \\ T & S \end{pmatrix} \begin{pmatrix} \Sigma_{0,pp} & \Sigma_{0,pa} \\ \Sigma_{0,ap} & \Sigma_{0,aa} \end{pmatrix} \begin{pmatrix} R & T^{\text{T}} \\ 0 & S \end{pmatrix} && \text{(47b)} \\ &= \begin{pmatrix} R\Sigma_{0,pp}R & R(\Sigma_{0,pa}S + \Sigma_{0,pp}T^{\text{T}}) \\ (T\Sigma_{0,pp} + S\Sigma_{0,ap})R & (T\Sigma_{0,pa} + S\Sigma_{0,aa})S + (T\Sigma_{0,pp} + S\Sigma_{0,ap})T^{\text{T}} \end{pmatrix} && \text{(47c)} \end{aligned}$$

Together, the non-Standardness of x_1 is given from Equation (8)

$$\begin{aligned} &\mathcal{S}(m_1, \Sigma_1) && \text{(48a)} \\ &= \tfrac{1}{2}\Big(\|m_1\|^2 + \text{tr}\,\Sigma_1 - D - \log\det\Sigma_1 \Big) && \text{(48b)} \\ &= \tfrac{1}{2}\Big(\|Am_0 + b\|^2 + \text{tr}(R^2\Sigma_{0,pp}) + \text{tr}(T\Sigma_{0,pa}S) + \text{tr}(S^2\Sigma_{0,aa}) + \text{tr}(T\Sigma_{0,pp}T^{\text{T}}) && \text{(48c)} \\ &\qquad + \text{tr}(S\Sigma_{0,ap}T^{\text{T}}) - D - \log\det\Sigma_0 - \log\det R - \log\det S \Big). && \text{(48d)} \end{aligned}$$

To find the minimum of $\mathcal{S}(m_1, \Sigma_1)$, minimize the above over r, s, T and b:

$$\arg\min_{r,s,T,b} \mathcal{S}(m_1, \Sigma_1). \tag{49a}$$

It is easy to see that $b = -Am_0$ minimizes Equation (48) as in this case $m_1 = 0$.

At the minimum, we find for r:

$$0 = \frac{\partial \mathcal{S}(m_1, \Sigma_1)}{\partial r_m} = -\frac{1}{r_m} + r_m(\Sigma_{0,pp})_{mm}, \tag{50}$$

for some $m = 1, \ldots, D/2$. We read off that $r_m = (\Sigma_{0,pp})_{mm}^{-1/2}$. In matrix notation:

$$R = \text{Diag}(\Sigma_{0,pp})^{-1/2}. \tag{51}$$

For s, T, we find the system:

$$0 = \frac{\partial \mathcal{S}(m_1, \Sigma_1)}{\partial s_n} = -\frac{1}{s_n} + \sum_{j=1}^{D/2} T_{nj}(\Sigma_{0,pa})_{jn} + s_n(\Sigma_{0,ap})_{nn} \tag{52a}$$

$$0 = \frac{\partial \mathcal{S}(m_1, \Sigma_1)}{\partial T_{op}} = s_p(\Sigma_{0,pa})_{po} + \sum_{k=1}^{D/2} T_{pk}(\Sigma_{0,aa})_{ko}. \tag{52b}$$

Multiplying the first equation by s_n, we find in matrix notation:

$$\begin{aligned} I &= \text{Diag}(T\Sigma_{0,pa}S + S^2\Sigma_{0,aa}) && \text{(53a)} \\ 0 &= S\Sigma_{0,pa} + T\Sigma_{0,pp}. && \text{(53b)} \end{aligned}$$

We solve the second equation for T (we use that $\Sigma_{0,pp}$ is invertible as it is positive definite):

$$T = -S\Sigma_{0,ap}\Sigma_{0,pp}^{-1}, \tag{54}$$

and insert into the first:

$$\begin{aligned} I &= \mathrm{Diag}(-S\Sigma_{0,ap}\Sigma_{0,pp}^{-1}\Sigma_{0,pa}S + S^2\Sigma_{0,aa}) && \text{(55a)} \\ &= \mathrm{Diag}(-S^2\Sigma_{0,ap}\Sigma_{0,pp}^{-1}\Sigma_{0,pa} + S^2\Sigma_{0,aa}). && \text{(55b)} \end{aligned}$$

The last step is due to $\mathrm{Diag}(\cdot)$ linear and S diagonal. We read off that:

$$S = \mathrm{Diag}(\Sigma_{0,aa} - \Sigma_{0,ap}\Sigma_{0,pp}^{-1}\Sigma_{0,pa})^{-1/2}. \tag{56}$$

Alternative solutions with negative signs are discarded by convention (without an effect on the covariance).

Inserting into Equation (47c), we find:

$$\begin{aligned} \Sigma_{1,pp} &= R\Sigma_{0,pp}R = \mathrm{Diag}(\Sigma_{0,pp})^{-1/2}\Sigma_{0,pp}\,\mathrm{Diag}(\Sigma_{0,pp})^{-1/2} = M(\Sigma_{0,pp}), && \text{(57a)} \\ \Sigma_{1,pa} &= R(\Sigma_{0,pa}S + \Sigma_{0,pp}T^{\mathrm{T}}) = R(\Sigma_{0,pa}S - \Sigma_{0,pp}\Sigma_{0,pp}^{-1}\Sigma_{0,pa}S) = 0, && \text{(57b)} \\ \Sigma_{1,ap} &= \Sigma_{1,pa}^{\mathrm{T}} = 0, && \text{(57c)} \\ \Sigma_{1,aa} &= (T\Sigma_{0,pa} + S\Sigma_{0,aa})S + (T\Sigma_{0,pp} + S\Sigma_{0,ap})T^{\mathrm{T}} && \text{(57d)} \\ &= (-S\Sigma_{0,ap}\Sigma_{0,pp}^{-1}\Sigma_{0,pa} + S\Sigma_{0,aa})S + (S\Sigma_{0,ap}\Sigma_{0,pp}^{-1}\Sigma_{0,pp} - S\Sigma_{0,ap})\Sigma_{0,pp}^{-1}\Sigma_{0,pa}S && \text{(57e)} \\ &= S(\Sigma_{0,aa} - \Sigma_{0,ap}\Sigma_{0,pp}^{-1}\Sigma_{0,pa})S = M(\Sigma_{0,aa} - \Sigma_{0,ap}\Sigma_{0,pp}^{-1}\Sigma_{0,pa}). && \text{(57f)} \end{aligned}$$

This concludes the proof:

$$m_1 = 0, \qquad \Sigma_1 = \begin{pmatrix} M(\Sigma_{0,pp}) & 0 \\ 0 & M(\Sigma_{0,aa} - \Sigma_{ap}\Sigma_{pp}^{-1}\Sigma_{pa}) \end{pmatrix}. \tag{58}$$

□

B.3 Proof of Theorem 1

The following statement will help us along the way:

Lemma 2. *For $A \in \mathbb{C}^D$, $Q \in \{O(D), U(D)\}$ with the corresponding Haar measure $p(Q)$:*

$$\mathbb{E}_{Q\in p(Q)}[(QAQ^*)_{ii}] = \frac{1}{D}\,\mathrm{tr}(A). \tag{59}$$

Proof. By symmetry, $\mathbb{E}_Q[(QAQ^*)_{11}] = \mathbb{E}_Q[(QAQ^*)_{ii}]$ for $i = 1, \ldots, D$. Thus, $\mathbb{E}_Q[(QAQ^*)_{11}] = \frac{1}{D}\sum_{i=1}^{D}\mathbb{E}_Q[(QAQ^*)_{ii}] = \frac{1}{D}\mathbb{E}_Q[\mathrm{tr}(QAQ^*)] = \frac{1}{D}\,\mathrm{tr}\,A$. □

When we write Q^*, we mean conjugate transpose if Q is sampled from the unitary group $U(D)$, and transpose if Q is from the orthogonal group $O(D)$. Whenever we only consider orthogonal Q, we will resort back to writing Q^{T}.

This allows us to directly estimate $\mathbb{E}_{Q\sim p(Q)}[\log\det M_p^2]$:

Lemma 3. *With the definitions in Section 5.1, $p(Q)$ either the Haar measure of orthogonal or unitary matrices, and Assumption 2. Then:*

$$\mathbb{E}_{Q\sim p(Q)}[\log\det M_p^2] \geq 0. \tag{60}$$

Proof. M_p is given by:

$$M_p^2 = \mathrm{Diag}(\Sigma_{0,pp})^{-1}. \tag{61}$$

The corresponding expectation value can be estimated via Jensen's inequality:

$$\mathbb{E}_{Q\sim p(Q)}[\log\det M_p^2] = \mathbb{E}_{Q\sim p(Q)}[\log\det \mathrm{Diag}(\Sigma_{0,pp})^{-1}] \tag{62a}$$

$$= -\mathbb{E}_{Q\sim p(Q)}[\log\prod_{i=1}^{D/2}(\Sigma_{0,pp})_{ii}] = -\sum_{i=1}^{D/2}\mathbb{E}_{Q\sim p(Q)}[\log(\Sigma_{0,pp})_{ii}] \tag{62b}$$

$$\geq -\sum_{i=1}^{D/2}\log\mathbb{E}_{Q\sim p(Q)}[(\Sigma_{0,pp})_{ii}] = -\frac{D}{2}\log\mathrm{tr}\,\Sigma/D \tag{62c}$$

$$= 0. \tag{62d}$$

By Assumption 2, $\mathrm{tr}\,\Sigma = D$. We have used Lemma 2 for evaluating $\mathbb{E}_{Q\sim p(Q)}[(\Sigma_{0,pp})_{ii}]$. □

As mentioned in Section 5.1, the main difficulty in estimating $\mathbb{E}_{Q\sim p(Q)}[\mathcal{S}(\Sigma_1(Q))]$ lies in $\mathbb{E}_{Q\sim p(Q)}[\log\det M_a^2]$. The following subsections show a path to do so.

B.3.1 Problem reformulation

In a first step, we reformulate this expectation so that it can be computed with the help of projected orbital measures [37].

We split the expectation over the Haar measure $p(Q)$ in two parts: One that defines which eigenvalues the $(D/2)\times(D/2)$ block $\Sigma_{0,aa}$ has (denote this as Q_{ap}) and, conditioned on this, another which rotates $\Sigma_{0,aa}$ into all possibles bases (denote this as Q_a). Formally, write Q as:

$$Q = \begin{pmatrix} I & 0 \\ 0 & Q_a \end{pmatrix} Q_{ap}. \tag{63}$$

We will replace the Schur complement $\Sigma_{0,aa} - \Sigma_{0,ap}\Sigma_{0,pp}^{-1}\Sigma_{0,pa}$ appearing in Proposition 2 by the corresponding block of the precision matrix $P_0 := \Sigma_0^{-1} = (Q\Sigma^{-1}Q^*)^{-1} = Q\Sigma^{-1}Q^*$ (e.g. [36, Section (0.7.3)]:

$$(P_{0,aa})^{-1} = ((\Sigma_0^{-1})_{aa})^{-1} = \Sigma_{0,aa} - \Sigma_{0,ap}\Sigma_{0,pp}^{-1}\Sigma_{0,pa}. \tag{64}$$

We give more details in the proof of the following lemma, which formalizes this step:

Lemma 4. *Given the definitions in Section 5.1 and Assumption 2. It holds that*

$$\mathbb{E}_{Q\sim p(Q)}[\log\det M_a^2] \geq -\sum_{i=1}^{D/2}\log\mathbb{E}_{Q_a\sim p(Q_a|Q_{ap})}[((P_{0,aa})^{-1})_{ii}]. \tag{65}$$

Proof. By Proposition 2, M_a^2 is given by:

$$M_a^2 = \mathrm{Diag}(\Sigma_{0,aa} - \Sigma_{0,ap}\Sigma_{0,pp}^{-1}\Sigma_{0,pa})^{-1}. \tag{66}$$

Being diagonal, its determinant is given by the product of its diagonal entries:

$$\mathbb{E}_{Q\sim p(Q)}[\log\det M_a^2] = \mathbb{E}_{Q\sim p(Q)}[\log\prod_{i=1}^{D/2}(\Sigma_{0,aa} - \Sigma_{0,ap}\Sigma_{0,pp}^{-1}\Sigma_{0,pa})_{ii}^{-1}] \tag{67a}$$

$$= \sum_{i=1}^{D/2}\mathbb{E}_{Q\sim p(Q)}[\log((\Sigma_{0,aa} - \Sigma_{0,ap}\Sigma_{0,pp}^{-1}\Sigma_{0,pa})_{ii}^{-1})] \tag{67b}$$

$$= -\sum_{i=1}^{D/2}\mathbb{E}_{Q\sim p(Q)}[\log((\Sigma_{0,aa} - \Sigma_{0,ap}\Sigma_{0,pp}^{-1}\Sigma_{0,pa})_{ii})]. \tag{67c}$$

Evaluating this expression is hard mainly because the $\Sigma_{0,ap}\Sigma_{0,pp}^{-1}\Sigma_{0,pa}$ involves the inverse of $\Sigma_{0,pp} = (Q\Sigma Q^*)_{pp}$, which depends on Q.

To circumvent this, note the following property of any nonsingular matrix M [36, Section (0.7.3)]. Split M into blocks as:

$$M = \begin{pmatrix} A & B \\ B^* & C \end{pmatrix}, \tag{68}$$

and do the same for its inverse:

$$M^{-1} = \begin{pmatrix} A' & B' \\ B'^* & C' \end{pmatrix} \tag{69}$$

Then, $(A')^{-1} = A - BC^{-1}B^*$, which is called the Schur complement M/C. This means we can rewrite

$$\Sigma_{0,aa} - \Sigma_{0,ap}\Sigma_{0,pp}^{-1}\Sigma_{0,pa} = (P_{0,aa})^{-1}, \tag{70}$$

where $P_0 = \Sigma_0^{-1}$ is the *precision matrix* of the rotated data. Given a rotation Q, it can easily be obtained from the precision matrix of the data in its original rotation:

$$\Sigma_0 = Q\Sigma Q^*, \qquad P_0 = Q\Sigma^{-1}Q^*. \tag{71}$$

Inserting this, we find the expectation value:

$$\mathbb{E}_{Q\sim p(Q)}[\log\det M_a^2] = -\sum_{i=1}^{D/2}\mathbb{E}_{Q\sim p(Q)}[\log(((P_{0,aa})^{-1})_{ii})]. \tag{72}$$

The logarithm can be drawn out via Jensen's inequality:

$$-\sum_{i=1}^{D/2}\mathbb{E}_{Q\sim p(Q)}[\log(((P_{0,aa})^{-1})_{ii})] \geq -\sum_{i=1}^{D/2}\log(\mathbb{E}_{Q\sim p(Q)}[((P_{0,aa})^{-1})_{ii}]). \tag{73}$$

This concludes the statement. □

B.3.2 Projected orbit expectation

The theory of projected orbital measures describes the distribution of eigenvalues of a randomly projected submatrix of some given matrix. Let us formalize this:

Fix a diagonal matrix $A = \mathrm{Diag}(a_1, \ldots, a_N)$. Then, then the *orbit* of A is defined as:

$$\mathcal{O}_A := \{QAQ^* : Q \in U(D)\}. \tag{74}$$

(The same definition also exists for orthogonal $Q \in O(D)$, but we keep it to the level we require here). All matrices in the orbit of $\mathcal{O}_A$ have the same eigenvalues.

The natural measure (probability distribution) on the orbit $\mathcal{O}_A$ is given by the image of the Haar measure on the unitary group $U(D)$. This can be thought of as the uniform measure on the group of unitary rotations. We call this measure the *orbital measure*.

We now cut out the $K \times K$ top left corner out of every matrix in $\mathcal{O}_A$:

$$P_K\mathcal{O}_A := \{P_K Y : Y \in \mathcal{O}_A\}. \tag{75}$$

We call this the *projected orbit*. The matrix P_K projects a matrix to its upper left corner:

$$P_K = (I_K; 0_{K\times(N-K)}). \tag{76}$$

The distribution of matrices in the projected orbit $P_K\mathcal{O}_A$ induced by the orbital measure is denoted as the *projected orbital measure* $\mu_{A,K}$. We are now interested in the eigenvalues of matrices in the projected orbit $P_K\mathcal{O}_A$.

Let $\mathrm{spectrum}$ be the function that assigns a matrix $Y \in \mathbb{C}^{K\times K}$ its eigenvalues $y_1, \ldots, y_K$. We will make use of a result that gives the distribution of eigenvalues of matrices in the projected orbit $P_K\mathcal{O}_A$. This is called the *radial part of the projected orbital measure* and is denoted as $\nu_{A,K}(x_1, \ldots, x_K)$.

$$\nu_{A,K}(x_1, \ldots, x_K) = \mathbb{P}_{X\sim\mu_{A,K}}[\mathrm{spectrum}(X) = (x_1, \ldots, x_K)]. \tag{77}$$

In other words, $\nu_{A,K}(x_1, \ldots, x_K)$ gives the probability density that a random matrix from the projected orbit of A has exactly eigenvalues $(x_1, \ldots, x_K)$. Its functional form was shown by [37]:

Theorem 4 (Radial part of projected orbital measure [37])**.** *Fix $A = (a_1, \dots, a_D)$ with $a_1 < \cdots < a_D$. For any $K = 1, \dots, D-1$, the density of eigenvalues of*

$$\nu_{A,K}(x_1, \dots, x_K) = c_{D,K} \frac{V(x_1, \dots, x_K) \det[M(a_j; x_i, \dots, x_{D-K+i})]_{i,j=1}^K}{\prod_{j-i \geq D-K+1}(x_j - x_i)}. \tag{78}$$

Here, the constant is given by:

$$c_{D,K} = \prod_{i=1}^{K-1} \binom{D-K+i}{i}, \tag{79}$$

and $M(a; y_1, \dots, y_N)$ is the B-spline:

$$M(a; y_1, \dots, y_n) := (N-1) \sum_{i: y_i > a} \frac{(y_i - a)^{n-2}}{\prod_{r: r \neq i}(y_i - y_r)}, \tag{80}$$

and V is the Vandermonde polynomial:

$$V(y_1, \dots, y_n) = \prod_{i<j}(y_j - y_i). \tag{81}$$

We will make use of the following variant of the Vandermonde determinant where all powers greater or equal to some k are increased by one:

Lemma 5. *For all $n \in \mathbb{N}$, $k = 1, \dots, n-1$ and distinct $a_i, i = 1, \dots, n$:*

$$\det \begin{pmatrix} 1 & \cdots & a_1^{k-1} & a_1^{k+1} & \cdots & a_1^n \\ \vdots & & \vdots & \vdots & & \vdots \\ 1 & \cdots & a_n^{k-1} & a_n^{k+1} & \cdots & a_n^n \end{pmatrix} = V(a_1, \dots, a_n) e_{n-k}(a_1, \dots, a_n). \tag{82}$$

with the elementary symmetric polynomial e_K given by Equation (17)*.*

Lemma 6. *Fix $A = (a_1, \dots, a_N)$ with $a_1 < \cdots < a_N$. For any $K = 1, \dots, N-1$, it holds that:*

$$\mathbb{E}_{a_1, \dots, a_K \sim \nu_{A,K}(x_1, \dots, x_K)}[x_1^{-1} + \cdots + x_K^{-1}] \tag{83a}$$

$$= (N-K)(-1)^{N-K} \sum_{j=1}^N a_j^{N-K-1} \log(a_j) R(a_j; a_{\neq j}) e_{K-1}(a_{\neq j}). \tag{83b}$$

Here, R is defined in Equation (17)*.*

Proof. We use the Andreief identity in the form of [41, Lemma 2.1]

$$\mathbb{E}_{x_1, \dots, x_k \sim \nu_{A,K}}[\tfrac{1}{x_1} + \cdots + \tfrac{1}{x_k}] \tag{84a}$$

$$= Z^{-1} \int_{(\mathbb{R}^K)_+} (\tfrac{1}{x_1} + \cdots + \tfrac{1}{x_k}) \det(x_i^{j-1}) \det(M(x_i; a_j, \dots, a_{j+N-K}))\, \mathrm{d}x_1 \cdots \mathrm{d}x_k \tag{84b}$$

$$= Z^{-1} \sum_{k=1}^K \det \int_{\mathbb{R}} x^{-\delta_{jk}} x^{j-1} M(x; a_i, \dots, a_{i+N-K})\, \mathrm{d}x \tag{84c}$$

$$= Z^{-1} \det \int_{\mathbb{R}} x^{j-1-\delta_{j1}} M(x; a_i, \dots, a_{i+N-K})\, \mathrm{d}x \tag{84d}$$

$$= Z^{-1} \det \begin{cases} \mu_{-1}(a_i, \dots, a_{i+N-K}) & j = 1 \\ \mu_{j-1}(a_i, \dots, a_{i+N-K}) & j > 1 \end{cases} \tag{84e}$$

where $\mu_k(t_1, \dots, t_n)$ the kth moment of the B-spline with knots $t_1, \dots, t_n$:

$$\mu_k(t_1, \dots, t_n) = \int x^k M(x; t_1, \dots, t_n)\, \mathrm{d}x. \tag{85}$$

We can now make use of the Hermite–Genocchi formula [38, Proposition 6.3]:

$$\int f^{(n-1)}(x) M(x; t_1, \dots, t_n)\, \mathrm{d}x = (n-1)! f[t_1, \dots, t_n], \tag{86}$$

so we can rewrite

$$\mu_k(t_1, \dots, t_n) = f_k[t_1, \dots, t_n], \tag{87}$$

with

$$f_{-1}(x) = (n-1)x^{n-2} \log x, \tag{88a}$$

$$f_k(x) = \binom{n+k-1}{k}^{-1} x^{n+k-1}. \tag{88b}$$

Together, we find

$$\mathbb{E}_{x_1,\dots,x_k \sim \nu_{A,K}}[\tfrac{1}{x_1} + \dots + \tfrac{1}{x_k}] = Z^{-1} \det(f_{i-1-\delta_{1i}}[a_j, \dots, a_{j+N-K}]). \tag{89}$$

The right hand side can be identified with the right hand side of [38, Proposition 6.4]. It is equal to:

$$Z^{-1} \det(f_i[a_j, \dots, a_{j+N-K}]) \tag{90a}$$

$$= Z^{-1} \left(\prod_{0<j-i\le N-K} (a_j - a_i)^{-1} \right) \begin{vmatrix} 1 & \cdots & 1 \\ a_1 & \cdots & a_N \\ \vdots & & \vdots \\ a_1^{N-K-1} & \cdots & a_N^{N-K-1} \\ f_1(a_1) & \cdots & f_1(a_N) \\ \vdots & & \vdots \\ f_K(a_1) & \cdots & f_K(a_N) \end{vmatrix} \tag{90b}$$

$$= Z^{-1} \left(\prod_{0<j-i\le N-K} (a_j - a_i)^{-1} \prod_{k=1}^{K} \binom{N-K+k}{k}^{-1} \right) (N-K) \begin{vmatrix} 1 & \cdots & 1 \\ a_1 & \cdots & a_N \\ \vdots & & \vdots \\ a_1^{N-K-1} & \cdots & a_N^{N-K-1} \\ a_1^{N-K-1} \log a_1 & \cdots & a_N^{N-K-1} \log a_N \\ a_1^{N-K+1} & \cdots & a_N^{N-K+1} \\ \vdots & & \vdots \\ a_1^{N-1} & \cdots & a_N^{N-1} \end{vmatrix} \tag{90c}$$

$$=: C_2 \det(M_{ij}) \tag{90d}$$

Here, C_2 reduces to:

$$C_2 = \frac{N-K}{V(a_1, \dots a_n)}. \tag{91a}$$

Then, the determinant of M_{ij} reads:

$$\det M_{ij} = \sum_{j=1}^{N} (-1)^{N-K+1+j} a_j^{N-K-1} \log(a_j) V(a_{\neq j}) \sum_{\substack{i_1 < \dots < i_{K-1} \\ i_\dots \neq j}} a_{i_1} \cdots a_{i_{K-1}} \tag{92a}$$

$$= V(a)(-1)^{N-K} \sum_j a_j^{N-K-1} \log(a_j) R(a_j; a_{\neq j}) e_{K-1}(a_{\neq j}), \tag{92b}$$

where $R(a_j; a_{\neq j})$ collects all the terms in $V(a)$ that were not contained in $V(a_{\neq j})$ up to sign:

$$R(a_j; a_{\neq j}) = \prod_{\substack{i=1 \\ i\neq j}}^{n} \frac{1}{a_i - a_j} = (-1)^{j-1} V(a_{\neq j})/V(a). \tag{93}$$

Note that the sign of $R(a_j; a_{\neq j})$ flips from $j \to j+1$, so the alternating nature of the above series remains.

Together, the desired expectation value reads:

$$\mathbb{E}_{x_1,\dots,x_k \sim \nu_{A,K}}[\tfrac{1}{x_1} + \dots + \tfrac{1}{x_k}] \tag{94a}$$

$$= (N-K)(-1)^{N-K} \sum_{j=1}^{N} a_j^{N-K-1} \log(a_j) R(a_j; a_{\neq j}) e_{K-1}(a_{\neq j}), \tag{94b}$$

which concludes the proof. □

We now connect this result to our situation. This paves the path from the reformulation in Lemma 4 to Theorem 1.

Corollary 1. *For the definitions in Section 5.1 and when Assumptions 3 and 4 are fulfilled, it holds that:*

$$\mathbb{E}_{Q\sim p(Q)}[\log\det M_a^2] \geq \tfrac{D}{2} \log\left((-1)^{\frac{D}{2}+1} \sum_{i=1}^{D} \lambda_i^{1-\frac{D}{2}} \log(\lambda_i) R(\lambda_i^{-1}; \lambda_{\neq i}^{-1}) e_{\frac{D}{2}-1}(\lambda_{\neq i}^{-1})\right). \tag{95}$$

Proof. Then, Lemma 2 tells us how to integrate over Q_a.

$$\mathbb{E}_{Q_a\sim p(Q_a|Q_{ap})}[((P_{0,aa})^{-1})_{ii}] = \mathrm{tr}((Q_{ap} P Q_{ap}^*)^{-1}) = \sum_{i=1}^{D/2} a_i(Q_{ap})^{-1}. \tag{96}$$

Here, we denote by $a_i(Q_{ap})$ the ith eigenvalue of $P_0 = Q_{ap} P Q_{ap}^*$, which depends on the "outer" rotation Q_{ap}.

We substitute the expectation over Q_{ap} with an expectation over the projected eigenvalues of the rotated precision matrix P_0:

$$\mathbb{E}_{Q_{ap}\sim p(Q)}[\mathbb{E}_{Q_a\sim p(Q_a|Q_{ap})}[((P_{0,aa})^{-1})_{ii}] = \mathrm{tr}((Q_{ap} P Q_{ap}^*)^{-1})] \tag{97a}$$

$$= \mathbb{E}_{a_1,\dots,a_{D/2}\sim \nu_{A,D/2}(a_1,\dots,a_{D/2}|\lambda_1^{-1},\dots,\lambda_D^{-1})}[a_1^{-1} + \dots + a_{D/2}^{-1}]. \tag{97b}$$

Here $X = (\lambda_1^{-1}, \dots \lambda_D^{-1})$ contains the eigenvalues of the precision matrix P, the inverse of the covariance Σ. Lemma 6 with $K = D/2$ tells us how to evaluate the above expression. Insert the result into Lemma 4 to obtain the result. □

B.3.3 Summary

We can now collect the above pieces to build the **proof of Theorem 1**:

Proof. Equation (15) is the version of the non-Standardness after a single layer when Assumptions 1 and 2 are fulfilled. Insert Lemma 3 (passive part) and Corollary 1 (active part) to obtain the result. The former required Assumption 2 and the latter Assumptions 3 and 4 to hold. □

B.3.4 Handling of imaginary part

If we allow for unitary rotations $Q \in U(D)$, real-valued data is typically rotated into imaginary space. In fact, the case that the input remains real even has probability zero:

$$\mathbb{P}[Qx \in \mathbb{R}^D] = 0. \tag{98}$$

This does not pose a problem for our theory: The covariance matrix is positive definite also for complex data and so it has a positive determinant and trace, which are the only quantities entering the non-Standardness $\mathcal{S}$ (see Equation (8)).

B.4 Proof of Theorem 2

Lemma 7. *With the definitions in Section 5.1 and $p(Q)$ the Haar measure over the orthogonal group $O(D)$:*

$$\mathbb{E}_{Q\sim p(Q)}[\log\det M_a^2] \geq D/2\log\left(1 - \frac{DD}{2(D+1)(D-1)}\frac{\mathrm{Var}[\lambda]}{\lambda_{\max}}\right). \tag{99}$$

Proof. By Proposition 2, M_a^2 is given by:

$$M_a^2 = \mathrm{Diag}(\Sigma_{0,aa} - \Sigma_{0,ap}\Sigma_{0,pp}^{-1}\Sigma_{0,pa})^{-1}. \tag{100}$$

The determinant of a diagonal matrix is equal to the product of the entries on the diagonal. By the permutation symmetry of $p(Q)$, we can pick the entry in the upper left corner:

$$\mathbb{E}_{Q\sim p(Q)}[\log\det M_a^2] = D/2\mathbb{E}_{Q\sim p(Q)}[\log((M_a^{-2})_{11})] \leq -D/2\log\mathbb{E}_{Q\sim p(Q)}[(M_a^2)_{11}]. \tag{101}$$

The last step is due to the Jensen inequality.

We are left with computing $\mathbb{E}_{Q\sim p(Q)}[(M_a^2)_{11}]$:

$$\begin{aligned}
\mathbb{E}_{Q\sim p(Q)}[(M_a^2)_{11}] &= \mathbb{E}_{Q\sim p(Q)}[(\Sigma_{0,aa} - \Sigma_{0,ap}\Sigma_{0,pp}^{-1}\Sigma_{0,pa})_{11}] && \text{(102a)}\\
&= \mathbb{E}_{Q\sim p(Q)}[(\Sigma_{0,aa})_{11}] - \mathbb{E}_{Q\sim p(Q)}[(\Sigma_{0,ap}\Sigma_{0,pp}^{-1}\Sigma_{0,pa})_{11}] && \text{(102b)}\\
&= \frac{1}{D}\operatorname{tr}\Sigma_0 - \mathbb{E}_{Q\sim p(Q)}[(\Sigma_{0,ap}\Sigma_{0,pp}^{-1}\Sigma_{0,pa})_{11}]. && \text{(102c)}
\end{aligned}$$

The first expectation can be exactly computed via Lemma 2.

The average trace of the second matrix is not so easy to evaluate. As $\Sigma_{0,pp}^{-1}$ is positive definite, we can replace it with the worst case in the expectation:

$$(\Sigma_{0,ap}\Sigma_{0,pp}^{-1}\Sigma_{0,pa})_{11} \geq (\Sigma_{0,ap}\lambda_{\max}^{-1}I\Sigma_{0,pa})_{11} = (\Sigma_{0,ap}\Sigma_{0,pa})_{11}\lambda_{\max}^{-1}. \tag{103}$$

$\lambda_{\max}$ is the largest eigenvalue of Σ, which does not depend on Q.

The expectation value can now be computed exactly:

$$\begin{aligned}
\mathbb{E}_{Q\sim p(Q)}[(\Sigma_{0,ap}\Sigma_{0,pa})_{11}] &= \sum_{i=1}^{D/2}\mathbb{E}_{Q\sim p(Q)}[(\Sigma_{0,ap})_{1i}(\Sigma_{0,pa})_{i1}] && \text{(104a)}\\
&= D/2\mathbb{E}_{Q\sim p(Q)}[(\Sigma_{0,ap})_{11}^2]. && \text{(104b)}
\end{aligned}$$

The last step is because each summand will have the same contribution. Writing the matrix multiplication out explicitly:

$$(\Sigma_{0,ap})_{11}^2 = (Q\,\mathrm{Diag}(\lambda_1,\ldots,\lambda_D)Q^*)_{11}^2 = (\sum_{j=1}^{D}Q_{1j}\lambda_jQ_{(D/2+1)j})^2 \tag{105}$$

Again by symmetry, we can exchange axes and write 2 instead of $D/2+1$ in what follows:

$$(\sum_{j=1}^{D}Q_{1j}\lambda_jQ_{(D/2+1)j})^2 = (\sum_{j=1}^{D}Q_{1j}\lambda_jQ_{2j})^2 = \sum_{j,k=1}^{D}\lambda_j\lambda_kQ_{1j}Q_{2j}Q_{1k}Q_{2k}. \tag{106}$$

Taking the expectation, we use the linearity of the expectation and are left with the following monomials of elements of Q:

1. $j=k$: $\mathbb{E}_{Q\sim p(Q)}[Q_{1j}^2Q_{2j}^2] = \mathbb{E}_{Q\sim p(Q)}[Q_{11}^2Q_{21}^2]$ as we can exchanges axes,
2. $j\neq k$: $\mathbb{E}_{Q\sim p(Q)}[Q_{1j}Q_{2j}Q_{1k}Q_{2k}] = \mathbb{E}_{Q\sim p(Q)}[Q_{11}Q_{21}Q_{12}Q_{22}]$ as we can exchange axes.

By [39], these amount to the following integrals of monomials of entries of random orthogonal matrices and the corresponding values:

$$1.\ \left\langle 2 \quad 2 \right\rangle = \frac{1}{D(D+2)}, \tag{107a}$$

$$2.\ \left\langle \begin{matrix} 1 & 1 \\ 1 & 1 \end{matrix} \right\rangle = -\frac{1}{D(D-1)(D+2)}. \tag{107b}$$

Together, we find

$$\mathbb{E}_{Q\sim p(Q)}[(M_a^2)_{11}] = 1 - \frac{1}{2(D+2)\lambda_{\max}} \left(\sum_{j=1}^{D} \lambda_j^2 - \frac{1}{D-1} \sum_{j\neq k} \lambda_j \lambda_k \right) \tag{108a}$$

$$= 1 - \frac{D^2}{2(D-1)(D+2)} \frac{\mathrm{Var}[\lambda]}{\lambda_{\max}}. \tag{108b}$$

Here, $\mathrm{Var}[\lambda] = \frac{1}{D}\operatorname{tr}\Sigma^2 - (\frac{1}{D}\operatorname{tr}\Sigma)^2$ is the variance of the eigenvalues of Σ.

Insert this to obtain the result. □

Lemma 8. *With the definitions in Section 5.1:*

$$\mathbb{E}_{Q\sim p(Q)}[\log\det M_a^2] \geq D/2 \log\left(1 - \frac{DD}{2(D+1)(D-1)} \frac{\mathrm{Var}[\lambda]}{\lambda_{\max}}\right). \tag{109}$$

Proof. The idea is to lower bound

$$\frac{\mathrm{Var}[\lambda]}{\lambda_{\max}} \tag{110}$$

by some function of L. We make use of following arithmetic mean-geometric mean (AM-GM) inequality by [40]:

$$\frac{\mathrm{Var}[\lambda]}{2\lambda_{\max}} \leq \bar{\lambda} - g \leq \frac{\mathrm{Var}[\lambda]}{2\lambda_{\min}}, \tag{111}$$

where g is the geometric mean of the eigenvalues:

$$g := \left(\prod_{i=1}^{D} \lambda_i \right)^{1/D}. \tag{112}$$

We can write the loss L directly via g and vice versa:

$$L = -\frac{1}{2}\log g^D = -\frac{D}{2}\log g, \tag{113a}$$

$$g = \exp(-2L/D). \tag{113b}$$

Rewrite Equation (111) to our needs:

$$\frac{\mathrm{Var}[\lambda]}{\lambda_{\max}} = \frac{\mathrm{Var}[\lambda]\lambda_{\min}}{\lambda_{\max}\lambda_{\min}} = \frac{2}{\kappa}\frac{\mathrm{Var}[\lambda]}{2\lambda_{\min}} \geq \frac{2}{\kappa}(1-g), \tag{114}$$

with κ the condition number of the covariance Σ.

As we want a bound that merely depends on the loss, we upper bound κ using a function of the loss, yielding a lower bound on $\mathrm{Var}[\lambda]/\lambda_{\max}$ that merely depends on the loss. The maximum of the condition value is given by:

$$\max_{\substack{\lambda_1,\ldots,\lambda_D \\ \sum_i \lambda_i = D \\ \prod_i \lambda_i^{1/D} = g}} \kappa = \frac{1+\sqrt{1-g^D}}{1-\sqrt{1-g^D}}. \tag{115}$$

This yields the required lower bound:

$$\frac{\mathrm{Var}[\lambda]}{\lambda_{\max}} \geq 2\frac{1-\sqrt{1-g^D}}{1+\sqrt{1-g^D}}(1-g), \tag{116}$$

which results in an overall upper bound:

$$\mathbb{E}_{Q\in O(D)}[\mathcal{S}(\Sigma_1(Q))] \leq \mathcal{S}(\Sigma) + \frac{D}{4}\log\left(1 - \frac{D^2}{(D-1)(D+2)}\frac{1-\sqrt{1-g^D}}{1+\sqrt{1-g^D}}(1-g)\right). \quad (117)$$

Replacing the expression in Equation (113b) for g yields the statement. □

We summarize to obtain the **proof of Theorem 2**:

Proof. Equation (15) is the form of non-Standardness $\mathcal{S}(\Sigma_1)$ (see Equation (8)) we need to evaluate when Assumptions 1 and 2 hold. Into this equation, insert Lemma 3 together with Lemma 7 for the first bound. For the second bound, insert Lemmas 3 and 8. □

B.5 Proof of Theorem 3

Proof. The non-Standardness will not increase by the action of a single layer given in Proposition 2 (compare Equation (14)). This holds regardless of the rotations of the individual blocks $Q_1, \dots Q_L$, so $\mathcal{S}(\Sigma) = \mathcal{S}(\Sigma_0) \geq \mathcal{S}(\Sigma_1) \geq \dots \geq \mathcal{S}(\Sigma_L)$. It is easy to see that γ decreases as $\mathcal{S}$ decreases by using $\mathcal{S} > 0$ to check that

$$\frac{\partial\gamma}{\partial\mathcal{S}} > 0. \quad (118)$$

Together, we have:

$$\gamma\big(\mathcal{S}(\Sigma_{L-1})\big) \leq \dots \leq \gamma\big(\mathcal{S}(\Sigma_0)\big). \quad (119)$$

Rewrite Theorem 2 as follows:

$$\mathbb{E}_{Q\in O(D)}[\mathcal{S}(\Sigma_1(Q))] \leq \gamma\big(\mathcal{S}(\Sigma_0)\big)\mathcal{S}(\Sigma), \quad (120)$$

and apply repeatedly:

$$\begin{aligned}
\mathbb{E}_{Q_1,\dots,Q_L\in O(D)}[\mathcal{S}(\Sigma_L)] &\leq \mathbb{E}_{Q_1,\dots,Q_{L-1}\in O(D)}[\gamma(\mathcal{S}(\Sigma_{L-1}))\mathcal{S}(\Sigma_{L-1})] && (121a)\\
&\leq \gamma(\mathcal{S}(\Sigma))\mathbb{E}_{Q_1,\dots,Q_{L-1}\in O(D)}[\mathcal{S}(\Sigma_{L-1})] && (121b)\\
&\leq \dots \leq \gamma(\mathcal{S}(\Sigma))^L\mathcal{S}(\Sigma_0) && (121c)
\end{aligned}$$

This shows the statement. □

C Compatible coupling architectures

All statements in this paper apply to the following architectures, where we assume each layer to be equipped with ActNorm [6]. To shorten the notation, we consider how a single dimension is transformed and rewrite the dependence on p_0 via a parameter vector $\theta = \theta(p_0)$, which is usually computed by a feed-forward neural network:

$$y = c(x;\theta), \quad (122)$$

short for:

$$(a_1)_i = c\big((a_0)_i;\theta_i(p_0)\big). \quad (123)$$

- **Affine coupling flows** in the form of NICE [4], RealNVP [5] and GLOW [6]:

  $$c(x;\theta) = sx + t. \quad (124)$$

  Here, $\theta = [s;t] \in \mathbb{R}_+ \times \mathbb{R}$.
- **Nonlinear squared flow** [19]:

  $$c(x;\theta) = ax + b + \frac{c}{1+(dx+h)^2}, \quad (125)$$

  for $\theta = [a,b,c,d,h] \in \mathbb{R}^5$.

- **SOS polynomial flows** [27]:

$$c(x;\theta) = \int_0^x \sum_{\kappa=1}^{k} \left(\sum_{l=0}^{r} a_{l,\kappa} u^l \right)^2 \mathrm{d}u + t. \tag{126}$$

  Here, $\theta = [t; (a_{l,\kappa})_{l,\kappa}] \in \mathbb{R} \times \mathbb{R}^{rk}$.

- **Flow++** [18]:

$$c(x;\theta) = s\sigma^{-1}\left(\sum_{j=1}^{K} \pi_j \sigma\left(\frac{x-\mu_j}{\sigma_j} \right) \right) + t. \tag{127}$$

  Here, $\theta = [s; t; (\pi_j, \mu_j, \sigma_j)_{j=1}^K] \in \mathbb{R}_+ \times \mathbb{R} \times (\mathbb{R} \times \mathbb{R} \times \mathbb{R}_+)^K$ and σ is the logistic function.

- **Spline flows** in the form of piecewise-linear, monotone quadratic [20], cubic [21], and rational quadratic [22] splines. Here, c is a spline of the corresponding type, parameterized by knots θ.

- **Neural autoregressive flow** [26] parameterize $c(x;\theta)$ by a feed-forward neural network, which can be shown to be bijective if all weights are positive and all activation functions are strictly monotone.

  One can also restrict the neural network $c(x;\theta)$ to have positive output and integrate it numerically. This was introduced as **unconstrained monotonic neural networks** [24].