DISENTANGLEMENT BY NONLINEAR ICA WITH GENERAL INCOMPRESSIBLE-FLOW NETWORKS (GIN)

Anonymous authors
Paper under double-blind review

ABSTRACT

A central question of representation learning asks under which conditions it is possible to reconstruct the true latent variables of an arbitrarily complex generative process. Recent breakthrough work by Khemakhem et al. (2019) on nonlinear ICA has answered this question for a broad class of *conditional* generative processes. We extend this important result in a direction relevant for application to real-world data. First, we generalize the theory to the case of unknown intrinsic problem dimension and prove that in some special (but not very restrictive) cases, informative latent variables will be automatically separated from noise by an estimating model. Furthermore, the recovered informative latent variables will be in one-to-one correspondence with the true latent variables of the generating process, up to a trivial component-wise transformation. Second, we introduce a modification of the RealNVP invertible neural network architecture (Dinh et al. (2016)) which is particularly suitable for this type of problem: the General Incompressible-flow Network (GIN). Experiments on artificial data and EMNIST demonstrate that theoretical predictions are indeed verified in practice. In particular, we provide a detailed set of exactly 22 informative latent variables extracted from EMNIST.

1 INTRODUCTION

Deep latent-variable models promise to unlock the key factors of variation within a dataset, opening a window to interpretation and granting the power to manipulate data in an intuitive fashion. The theory of identifiability in linear independent component analysis (ICA) (Comon (1994)) tells us when this is possible, if we restrict the model to a simple linear transformation, but until recently there was no corresponding theory for the highly nonlinear models needed to manipulate complex data. We can often successfully approximate the marginal distribution of observed data with such models, but there were no theoretical guarantees of them also being able to approximate the distribution of the generating latent space. This changed with the recent breakthrough work by Khemakhem et al. (2019), which showed that under relatively mild conditions, it is possible to recover the joint data and latent space distribution, up to a simple transformation in the latent space. The key requirement is that the generating process is conditioned on a variable which is observed along with the data. This condition could be a class label, time index of a time series, or any other piece of information related to the given data. They interpret their theory as a *nonlinear* version of ICA.

This work extends this theory in a direction relevant for application to real-world data. The existing theory assumes knowledge of the intrinsic problem dimension, but this is unrealistic for anything but artificially generated datasets. Here, we show that in the special case of Gaussian latent space distributions, the intrinsic problem dimension can be *discovered*. The important latent variables are organically separated from noise variables by the estimating model. Furthermore, the variables discovered correspond to the true generating latent variables, up a trivial component-wise translation and scaling. Very similar results exist for other members of the exponential family with two parameters, such as the beta and gamma distributions.

We introduce a variant of the RealNVP invertible neural network (Dinh et al. (2016)): the General Incompressible-flow Network (GIN). The flow is called *incompressible* in reference to fluid dynamics, since it preserves volumes: the Jacobian determinant is simply unity. We emphasise its *generality* and increased expressive power in comparison to previous volume-preserving flows, such as NICE (Dinh et al. (2014)). As already noted in Khemakhem et al. (2019), flow-based generative

models are a natural fit for the theory of nonlinear ICA, as are the variational autoencoders (VAEs) (Kingma & Welling (2013)) used in that work. For us, major advantages of an invertible architecture over VAEs are the ability to directly optimize the likelihood, and freedom from the requirement to specify the dimension of the model's latent space. An INN always has a latent space of the same dimension as the data. In addition, the forward and backward models share parameters, saving the effort of learning separate, but complementary models for each direction.

In summary, our work makes the following contributions:

- We extend the theory of nonlinear ICA to allow for unknown intrinsic problem dimension. Doing so, we find that this dimension can be discovered and a one-to-one correspondence between generating and estimated latent variables established.
- We propose as an implementation an invertible neural network obtained by modifying the RealNVP architecture. We call our new architecture GIN: the General Incompressible-flow Network.
- We demonstrate the viability of the model on artificial data and the EMNIST dataset. We extract exactly 22 meaningful variables from EMNIST, encoding both global and local features.

2 Related Work

The basic goals of nonlinear ICA stem from the original work on linear ICA. An influential formulation, as well as the first identifiability results, were given in Comon (1994). These stated the conditions which allow the generating latent variables to be discovered, when the mixing function is a linear transformation. However, it was shown in Hyvärinen & Pajunen (1999) that this identifiability does not extend to general nonlinear functions.

The first identifiability results in nonlinear ICA came in Hyvärinen & Morioka (2016) and Hyvärinen & Morioka (2017), applied to time series, and implemented via a discriminative model and semi-supervised learning. A more general formulation, valid for other forms of data, was given in Hyvärinen et al. (2018) and the theory was extended to generative models in Khemakhem et al. (2019), where experiments were implemented by a VAE.

Many authors have addressed the general problem of *disentanglement*, and proposed models to learn disentangled features. Prominent among these is β-VAE (Higgins et al. (2017)) and its variations (e.g. Chen et al. (2018)) which augment the standard ELBO loss with tunable hyperparameters to encourage disentanglement. There are also attempts to modify the GAN framework (Goodfellow et al. (2014)) such as InfoGAN (Chen et al. (2016)), which tries to maximize the mutual information between some dimensions of the latent space and the observed data. Many of these approaches are unsupervised. However, as pointed out and empirically demonstrated in Locatello et al. (2018), unsupervised models are in general unidentifiable. This is because they do not take into account conditioning in the generating latent space, made clear in Khemakhem et al. (2019).

The invertible neural networks in this work build upon the NICE framework (Dinh et al. (2014)) and its extension in RealNVP (Dinh et al. (2016)). A similar network design to ours is Ardizzone et al. (2019). This is a conditioned INN based on RealNVP, however the conditioning information is applied as a parameter of the network. The authors find in experiments with MNIST that meaningful variables are present in their latent space, but are rotated such that they are not aligned with the axes of the space. In this work, the conditioning is only present as a parameter of the latent space distributions. As a result, it is covered by the theory of Khemakhem et al. (2019) and its extension here, which results in non-rotated, meaningful latent variables.

3 Theory

3.1 Existing Theory of Nonlinear ICA

This section reviews the main results from Khemakhem et al. (2019). Suppose the existence of the following three random variables: a latent generating variable $\mathrm{z} \in \mathbb{R}^n$, a condition $\mathrm{u} \in \mathbb{R}^m$ and a data point $\mathrm{x} \in \mathbb{R}^d$, where $n \leq d$. The distribution of z is a factorial member of the exponential family with k sufficient statistics, conditioned on u. In its most general form the distribution can be

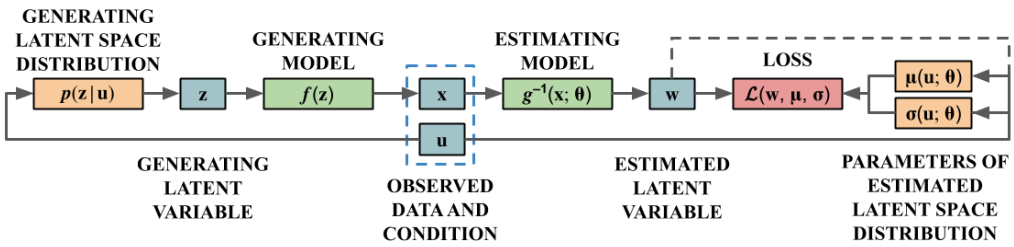


Figure 1: Relationship between the variables and functions defined in this section, as well as indication of the training scheme. The $\boldsymbol{\theta}$ are the weights parameterizing g which are optimized by gradient descent on the loss. This example is specifically for a Gaussian latent space, due to the parameters specified on the right. The dotted line from w to $\mu(\mathrm{u};\boldsymbol{\theta})$ and $\sigma(\mathrm{u};\boldsymbol{\theta})$ indicates that w may be used to update the parameters of the estimated latent space distribution, as described in Section 4.2 below.

written as

$$p_{\mathrm{z}}(\mathrm{z}|\mathrm{u}) = \prod_{i=1}^{n} \frac{Q_i(z_i)}{Z_i(\mathrm{u})} \exp\left[\sum_{j=1}^{k} T_{i,j}(z_i)\lambda_{i,j}(\mathrm{u})\right] \tag{1}$$

where the $T_{i,j}$ are the sufficient statistics, the $\lambda_{i,j}$ their coefficients and Z_i the normalizing constant. Q_i is called the base measure, which in many cases is simply 1. Explicitly applying (1) to a Gaussian with diagonal covariance:

$$p_{\mathrm{z}}(\mathrm{z}|\mathrm{u}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_i^2(\mathrm{u})}} \exp\left[-\frac{(\mathrm{z}-\mu_i(\mathrm{u}))^2}{2\sigma_i^2(\mathrm{u})}\right] \tag{2}$$

$$= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_i^2(\mathrm{u})}} \exp\left[-\frac{\mathrm{z}^2}{2\sigma_i^2(\mathrm{u})} + \frac{\mathrm{z}\cdot\mu_i(\mathrm{u})}{\sigma_i^2(\mathrm{u})} - \frac{\mu_i(\mathrm{u})^2}{2\sigma_i^2(\mathrm{u})}\right] \tag{3}$$

so we can write

$$T_{i,1}(z_i) = z_i, \quad T_{i,2}(z_i) = z_i^2, \quad \lambda_{i,1}(\mathrm{u}) = \frac{\mu_i(\mathrm{u})}{\sigma_i^2(\mathrm{u})} \quad \text{and} \quad \lambda_{i,2}(\mathrm{u}) = -\frac{1}{2\sigma_i^2(\mathrm{u})}. \tag{4}$$

The variable x is the result of an arbitrarily complex, injective transformation from the generating latent space to the data space: $\mathrm{x} = f(\mathrm{z})$ where f is not necessarily deterministic. In general, an observed dataset $\mathcal{D}$ will consist only of instances of x and u. The task of nonlinear ICA is to *unmix* or *disentangle* the data to recover the generating latent variables z, as well as the form of the function f and its inverse.

In addition to the above specification of z, u and x, some conditions are necessary to ensure the latent variables can be recovered. The most important of these concerns the variability of $\lambda_{i,j}(\mathrm{u})$ under u. However, as long as the $\lambda_{i,j}$ are randomly and independently generated, and there are at least $nk+1$ distinct conditions u, this condition is almost surely fulfilled. More details can be found in Khemakhem et al. (2019).

If the necessary conditions are met, the unmixing can be achieved with a sufficiently general, invertible function approximator g, for example a VAE or a flow-based model. A variable w sampled from the latent space of this model must follow a conditionally independent exponential probability distribution, conditioned on the same condition u as z:

$$p_{\mathrm{w}}(\mathrm{w}|\mathrm{u}) = \prod_{i=1}^{n} \frac{Q_i'(w_i)}{Z_i'(\mathrm{u})} \exp\left[\sum_{j=1}^{k} T_{i,j}'(w_i)\lambda_{i,j}'(\mathrm{u})\right]. \tag{5}$$

Note that k and n must be the same as for the unknown latent space. In the limit of infinite data and perfect convergence, the estimating model will be able to exactly reproduce the distribution of x: $p_{f,\mathrm{z}}(\mathrm{x}|\mathrm{u}) = p_{g,\mathrm{w}}(\mathrm{x}|\mathrm{u})$. In this case, the vector of sufficient statistics $\widetilde{\boldsymbol{T}}$ from the generating latent space will be related to that of the estimating latent space by an affine transformation:

$$\widetilde{\boldsymbol{T}}(\mathrm{z}) = \boldsymbol{A}\widetilde{\boldsymbol{T}}'(\mathrm{w}) + \boldsymbol{c} \tag{6}$$

where $\boldsymbol{A}$ is some constant, invertible, square matrix and $\boldsymbol{c}$ some constant vector. The relationship holds for all values of z and w. This is the main theoretical result of Khemakhem et al. (2019), which will be generalized in the following.

3.2 NONLINEAR ICA WITH UNKNOWN LATENT SPACE DIMENSION

The current theory of identifiability in nonlinear ICA assumes that the intrinsic dimension of the generating latent space is known. The dimension of the latent space of the estimating model is then assumed to be the same. When dealing with a real-world dataset, the dimension of the latent space is in general unknown. Here we show how to expand the theory to this case.

Suppose the dimension of the generating latent space is n and the dimension of the latent space used by the estimating model is n', where $n' \geq n$. In this case, under the same conditions required by the theorem that results in equation (6), we can derive a very similar relationship between variables in the generating and estimated latent spaces (see Appendix A). In brief, we maintain the relationship in equation (6), adjusting for the different dimensionality of z and w by making $\boldsymbol{A}$ into a rectangular matrix of size $nk \times n'k$. $\boldsymbol{A}$ is no longer necessarily square but maintains full rank (with rank nk). The number of sufficient statistics k per dimension must still be specified and cannot be inferred.

3.3 NONLINEAR ICA WITH A GAUSSIAN LATENT SPACE

Armed with knowledge of the relationship between elements of the true generating latent space and the latent space estimated by the model (equation (6)), we can ask whether any stronger results hold for particular special cases. We hope in particular to induce sparsity in the matrix $\boldsymbol{A}$, so that the estimated latent space is related to the true one by as simple a transformation as possible. For the case of a Gaussian latent space distribution, we do exactly that, demonstrating a result of equivalent strength to linear ICA. Refer to Appendix B for the proof.

Our result is that each generating latent variable z_i is related to exactly one estimated latent variable w_j, for some j, as

$$z_i = a_i w_j + b_i \tag{7}$$

for some constant scaling and translation a_i and b_i. Furthermore, each estimated latent variable w_j is related to at most one z_i. If the estimating latent space has higher dimension than the dimension of the generating latent space, some estimating latent variables are not related to any generating latent variables and so must encode only noise. This acts as a dimension discovery mechanism, since the estimating latent space organically splits into informative and non-informative parts, with the dimension of the informative part equal to the unknown intrinsic dimension of the generating latent space. Very similar results can be derived for all common continuous two-parameter members of the exponential family, such as the gamma and beta distributions. See Appendix C.

4 EXPERIMENTS

Experiments on artificial datasets confirm the theory for a normally distributed latent space, as well as identifying potential causes of failure. Experiments on the EMNIST dataset (Cohen et al., 2017) demonstrate the ability of GIN to estimate independent and interpretable latent variables from real-world data.

4.1 MODEL DESCRIPTION

The GIN model is similar in form to RealNVP (Dinh et al. (2016)) and shares its flexibility, but retains the volume-preserving properties of the NICE framework (Dinh et al. (2014)). Preserving volume creates a direct relationship between the standard deviation of the distribution of an estimated latent variable and its importance, as the standard deviation encodes the volume contributed by that variable, which is then preserved by the network.

RealNVP coupling layers split D-dimensional input x into two parts, $x_{1:d}$ and $x_{d+1:D}$ where $d < D$. The output of the layer is the concatentation of $y_{1:d}$ and $y_{d+1:D}$ with

$$y_{1:d} = x_{1:d} \tag{8}$$

$$y_{d+1:D} = x_{d+1:D} \odot \exp(s(x_{1:d})) + t(x_{1:d}) \tag{9}$$

where addition, multiplication and exponentiation are applied component-wise. The logarithm of the Jacobian determinant of a coupling layer is simply the sum of the scaling function $s(x_{1:d})$. Volume preservation is achieved by setting the final component $s(x_d)$ of s to the negative sum of the

previous components. Therefore the total sum of $s(x_{1:d})$ is zero, and the logarithm of the Jacobian determinant is also zero. Hence the Jacobian determinant is unity and volume is preserved. The network is free to allow the volume contribution of the outputs of any coupling layer to grow, but only by shrinking the other outputs in direct proportion. As well as enforcing a strong correspondence between the importance of a latent variable and its standard deviation, we believe volume-preservation has a regularizing effect, since it is a very strong constraint, comparable to orthogonality in linear transformations.

4.2 Optimization Method

The experiments in this section deal with labeled data, where each data point belongs to one of M different classes. This class label is used as the condition u associated with each data point x. In the estimated latent space, all data instances with the same label should belong to the same Gaussian distribution. Hence we are learning a Gaussian mixture in the estimated latent space, with M mixture components. Since the distribution for each class in the estimated latent space is required to be factorial, the variance of each mixture component is diagonal, and we can write $\sigma_i^2(\mathrm{u})$ for the variance in the i-th dimension.

Given a set of data, condition pairs $\mathcal{D} = \{(\mathrm{x}^{(1)}, \mathrm{u}^{(1)}), \ldots, (\mathrm{x}^{(N)}, \mathrm{u}^{(N)})\}$ and model g, parameterized by $\boldsymbol{\theta}$ (where g maps from the latent space to the data space) we can construct a loss from the log-likelihood. Using the change of variables formula, we have $\log p(\mathrm{x}|\mathrm{u}) = \log p(\mathrm{w}|\mathrm{u})$, with no Jacobian term since the transformation $\mathrm{w} = g^{-1}(\mathrm{x}; \boldsymbol{\theta})$ is volume-preserving. To maximize the likelihood of $\mathcal{D}$, we minimize the negative log-likelihood of w in the estimated latent space (refer to equation (2) for the likelihood of a Gaussian distribution):

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{\mathcal{D}}\left[\sum_{i=1}^{n}\left(\frac{[g_i^{-1}(\mathrm{x}; \boldsymbol{\theta}) - \mu_i(\mathrm{u}; \boldsymbol{\theta})]^2}{2\sigma_i^2(\mathrm{u}; \boldsymbol{\theta})} + \log(\sigma_i(\mathrm{u}; \boldsymbol{\theta}))\right)\right]. \tag{10}$$

Take $\mathcal{D}$ to be a mini-batch of data. The mean and variance of a mixture component is updated at each iteration as the mean and variance of the transformations to latent space of all data points in $\mathcal{D}$ belonging to that mixture component:

$$\mu_i(\mathrm{u}'; \boldsymbol{\theta}) \leftarrow \mathbb{E}_{\mathcal{D}:\mathrm{u}=\mathrm{u}'}(g_i^{-1}(\mathrm{x}; \boldsymbol{\theta})) \tag{11}$$

$$\sigma_i^2(\mathrm{u}'; \boldsymbol{\theta}) \leftarrow \mathrm{Var}_{\mathcal{D}:\mathrm{u}=\mathrm{u}'}(g_i^{-1}(\mathrm{x}; \boldsymbol{\theta})). \tag{12}$$

Hence the parameters of the mixture components change in each batch, according to the data present. The notation $\mu_i(\mathrm{u}; \boldsymbol{\theta})$ does not indicate that μ is directly parameterized by $\boldsymbol{\theta}$ and learned, instead it indicates that μ is a function of g^{-1} which is parameterized by $\boldsymbol{\theta}$. A change in $\boldsymbol{\theta}$ will also change the output of μ, given the same mini-batch of data $\mathcal{D}$. The same holds for $\sigma_i(\mathrm{u}; \boldsymbol{\theta})$.

4.3 Artificial Data

4.3.1 Experiment 1: conditions of theory fulfilled

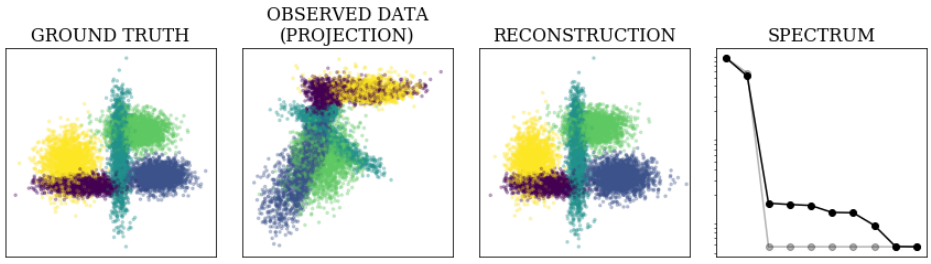


Figure 2: Successful reconstruction by GIN of the two informative latent variables out of ten in total. The other eight are correctly identified as noise. The observed data is ten-dimensional (projection into two dimensions shown here). The spectrum shows the standard deviation of each variable of the reconstruction (in black, log scale) which quantifies its importance. Ground truth is in gray. There is a clear distinction between the two informative dimensions and the noise dimensions, showing that GIN has correctly detected a two-dimensional manifold in the ten-dimensional data presented to it.

Samples are generated in two dimensions, conditioned on five different cluster labels. The means of the clusters are chosen independently from a uniform distribution on $[-5, 5]$ and variances from

a uniform distribution on $[.5, 3]$.[1] This data is then concatenated with independent Gaussian noise in eight dimensions to make a ten-dimensional generating latent space, where only the first two variables are informative. The noise is scaled by 0.01 to be small in comparison to the informative dimensions. The latent space samples are then passed through a RealNVP network with 8 fully connected coupling blocks with randomly initialized weights to produce the observed data. This acts as a highly nonlinear mixing which can only be successfully treated with nonlinear methods.

GIN is used as the estimating model, with 8 fully connected coupling blocks (full details in Appendix D). Training converges quickly and stably using the Adam optimizer (Kingma & Ba (2014)) with initial learning rate 10^{-2} and other values set to the usual recommendations. Batch size is 1,000 and the data is augmented with Gaussian noise ($\sigma = 0.01$) at each iteration. After convergence of the loss, the learning rate is reduced by a factor of 10 and trained again until convergence.

Over a number of experiments we made the following observations:

- The model converges stably and gives importance (quantified by standard deviation) to only two variables in the estimated latent space, provided there is sufficient overlap between the mixture components in the generating latent space.
- Where there is not enough overlap in the generating latent space, the model cannot recognize common variables across all the different classes, and tends to split one genuine dimension of variation into two or more in its estimated latent space. This appears to be a problem of finite data. We have observed that when this behaviour occurs, and if the gap between mixture components is not too large, it can be prevented by increasing the number of samples so that the space between the mixture components is better filled (see Figures 6 and 7 in Appendix E). This is consistent with the theory, where equation (6) is true asymptotically. Since the latent space distributions are members of the exponential family, they have support across the entire domain of the latent space, hence gaps can never remain in the limit of infinite samples.
- Choice of learning rate is important. If the initial learning rate is too low, training gets stuck in bad local optima, where single true variables are split into several latent dimensions.

4.3.2 Experiment 2: conditions of theory not fulfilled

Samples are generated as in Experiment 1, but with only three mixture components. Since there are two sufficient statistics per dimension, and two dimensions of variation, according to the theory we need at least $nk + 1 = 5$ distinct conditions u for equation (6) to hold (see section 3.1). Therefore, we might not expect successful experiments. Nonetheless, we observe essentially the same results as for the previous experiments (see Figure 8 in Appendix E), with the same caveats regarding gaps between the mixture components in the generating latent space. This suggests that the conditions derived in Khemakhem et al. (2019), although sufficient for disentanglement, are not necessary.

4.4 EMNIST

4.4.1 Experiment

The data comes from the EMNIST Digits training set of 240,000 images of handwritten digits with labels (Cohen et al. (2017)). EMNIST is a larger version of the well-known MNIST dataset, and also includes handwritten letters. Here we use only the digits. The digit label is used as the condition u, hence we construct a Gaussian mixture in the estimated latent space with 10 mixture components. According to the theory (see 3.1), these are only enough conditions to guarantee identifiability if there are only four informative latent variables in the generating latent space. We expect that the true number of informative variables is somewhat higher, so as in Experiment 2 above, we are operating outside of the guarantees of the theory. In addition, the true generative process of human handwriting may not exactly fulfill our method's assumptions, and we might still lack data on rare or subtle variations, despite the large size of the dataset in comparison to MNIST.

[1] These values are the same as in the first artificial data experiment in Khemakhem et al. (2019). The difference is that the data will be projected into a higher-dimensional space and that we will not assume that the estimating latent space is two-dimensional, this will instead be inferred by the model.

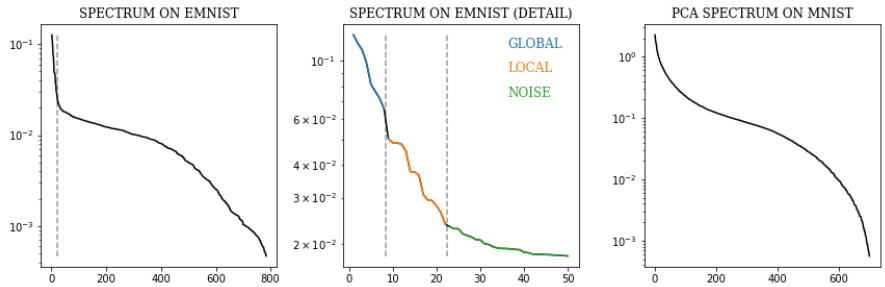
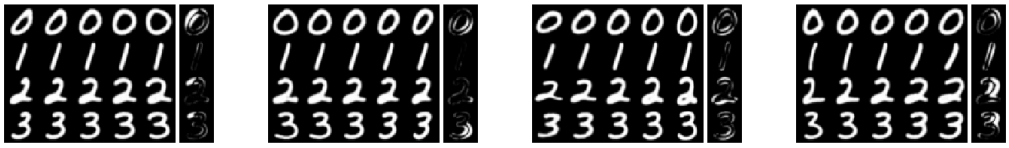


Figure 3: Spectrum of sorted standard deviations derived from training GIN on EMNIST. On the right is the equivalent spectrum from PCA, a linear method, on MNIST. The nonlinear spectrum exhibits a sharp knee not obtained by linear methods. In the nonlinear spectrum, the 22 most important latent variables (measured by the standard deviations of test data transformed to the latent space) encode information about the shape of a digit, while the rest of the latent variables encode noise. This distinction is marked with a dotted line in the left and center figures. Within the first 22 variables, the first eight encode global information, such as slant and width, whereas the following 14 encode more local information. This distinction is marked in the center figure only. Refer to Appendix F to see the effect of each of the first 22 latent variables on the digits.

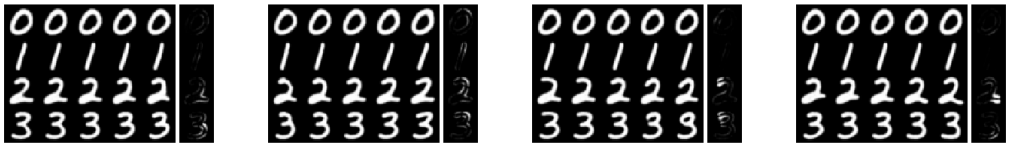
The estimating model is a GIN which uses convolutional coupling blocks and fully connected coupling blocks to transform the data to the latent space (full details in Appendix D). Optimization is with the Adam optimizer, with initial learning rate 3e-4. Batch size is 240 and the data is augmented with Gaussian noise ($\sigma = 0.01$) at each iteration. The model is trained for 45 epochs, then for a further 50 epochs with the learning rate reduced by a factor of 10.

4.4.2 RESULTS



(a) Variable 1 (b) Variable 8 (c) Variable 3 (d) Variable 4

Figure 4: Selection of global latent variables found by GIN. Each row is conditioned on a different digit label. The variable runs from -2 to +2 standard deviations across the columns, with all other variables held constant at their mean value. The rightmost column shows a heatmap of the image areas most affected by the variable, computed as the absolute pixel difference between -1 and +1 standard deviations. Variable 1 controls the width of the top half of a digit, whereas variable 8 controls the width of the bottom half. Width in both cases is somewhat entangled with slant. Variable 3 controls the height and variable 4 controls how bent the digit is. Full set of variables for all digits in Appendix F.



(a) Variable 11 (b) Variable 12 (c) Variable 13 (d) Variable 16

Figure 5: Selection of local latent variables found by GIN. Refer to Figure 4 above for explanation of the layout. Variable 11 controls the shape of curved lines in the middle right. Digits without such lines are not affected. Variable 12 controls extension towards the upper right. Variable 13 modifies the top left of 2, 3 and 7 only (7 not shown here) and variable 16 modifies only the lower right stroke of a 2. Full set of variables for all digits in Appendix F.

The model encodes information into 22 broadly interpretable latent space variables, with the remainder adding small amounts of noise. The effects of each of the interpretable variables can be seen in Appendix F. The eight variables with the highest standard deviation encode global information, such as slant, line thickness and width which affect all digits more or less equally. The remaining 14

meaningful variables encode more local information, which does not affect all digits equally. Some variables, particularly the global ones, are not entirely disentangled from another. Nevertheless, the results are compelling and suggest that the assumptions required by the theory are approximately met.

We observed some global features which are not usually seen in disentanglement experiments on MNIST (e.g. Chen et al. (2016), Dupont (2018)). These experiments usually obtain digit slant, width and line thickness as the major global independent variables. We too observe slant and line thickness as independent variables (variables 2 and 5), but find width to be split into two variables, one governing the width of the upper half of a digit (variable 1) and the other the lower half (variable 8). This makes sense, since these can in fact vary independently. We also observe the height of a digit (variable 3) (see Figure 4). This does not usually appear in disentanglement experiments, possibly because it is too subtle a variation for those experiments, but possibly because it is not present or not discoverable in the smaller MNIST dataset but is in EMNIST.

Local features are also usually not observed in such experiments, so the variables which control these are particularly interesting. These variables modify only a region of the digit, leaving the rest of the digit untouched. In addition, digits which do not have the feature which is being modified in that region are left alone. Examples include variable 13, which changes the orientation of the top-left stroke in 2, 3 and 7, and variable 16, which modifies only the lower-right stroke of a 2 (see Figure 5). The full set of local and global variables can be seen in Appendix F.

We can compare the latent space representations of digits from different classes by normalizing the latent space values with the mean and variance of the class the digit belongs to. To do so, we calculate the mean and variance in the latent space of some set of digits which all belong to the same class. We used a subset of the EMNIST test dataset of 40,000 images for this purpose. We can then transfer a normalized value into the distribution of another digit as a method of changing the conditioning while retaining information about style. Samples generated with different conditions (class labels) but a consistent style can be found in Appendix F.

4.4.3 Potential Improvements

Increasing the number of conditioning variables would bring this work closer in line with the existing theory. At the current value of ten conditioning variables, the theory only applies if there are only four informative generating latent variables, which is almost certainly too low a number. One option to increase the number of conditions is to relax the labels from hard to soft, i.e. compute posterior probabilities of class membership given a data example. This could be achieved by information distillation as in Hinton et al. (2015). By doing so, important variations within a class could be encoded, such as the crossbar on a 7. Although there are many examples in the dataset of the digit 7 with a crossbar, GIN has apparently treated it as an anomalous feature. As a result, samples never show a 7 with a crossbar (see Appendix F). With a soft label, a 7 with a crossbar could be grouped with similar examples, allowing the crossbar to be expressed when the appropriate condition is selected. However, allowing class labels to become continuous requires non-trivial modifications to the optimization method used in this work (Sec. 4.2).

5 Conclusion and Outlook

We have expanded the theory of nonlinear ICA to cover problems with unknown intrinsic dimension and demonstrated an implementation with GIN, a new volume-preserving modification of the RealNVP invertible network architecture. The variables discovered by GIN in EMNIST are interpretable and detailed enough to suggest that the assumptions made about the generating process are approximately true. Furthermore, our experiments with EMNIST demonstrate the viability of applying models inspired by the new theory of nonlinear ICA to real-world data, even when not all of the conditions of the theory are met.

It is not clear if the methods from this work will scale to larger problems. However, given the recent advances of similar flow-based generative models in density estimation on larger datasets such as CelebA and ImageNet (e.g. Kingma & Dhariwal (2018), Ho et al. (2019)), it is a plausible prospect. More investigation is necessary in this direction.

REFERENCES

Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Guided image generation with conditional invertible neural networks. *arXiv:1907.02392*, 2019.

Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pp. 2610–2620, 2018.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pp. 2172–2180, 2016.

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

Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv:1410.8516*, 2014.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. *arXiv:1605.08803*, 2016.

Emilien Dupont. Learning disentangled joint continuous and discrete representations. In *Advances in Neural Information Processing Systems*, pp. 710–720, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-VAE: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2(5):6, 2017.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, 2015.

Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. *arXiv:1902.00275*, 2019.

Aapo Hyvärinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ICA. In *Advances in Neural Information Processing Systems*, pp. 3765–3773, 2016.

Aapo Hyvärinen and Hiroshi Morioka. Nonlinear ICA of temporally dependent stationary sources. In *Proceedings of Machine Learning Research*, 2017.

Aapo Hyvärinen and Petteri Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439, 1999.

Aapo Hyvärinen, Hiroaki Sasaki, and Richard E Turner. Nonlinear ICA using auxiliary variables and generalized contrastive learning. *arXiv:1805.08651*, 2018.

Jörn-Henrik Jacobsen, Arnold Smeulders, and Edouard Oyallon. i-RevNet: Deep invertible networks. *arXiv:1802.07088*, 2018.

Ilyes Khemakhem, Diederik P Kingma, and Aapo Hyvärinen. Variational autoencoders and nonlinear ICA: A unifying framework. *arXiv:1907.04809*, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv:1312.6114*, 2013.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pp. 10215–10224, 2018.

Francesco Locatello, Stefan Bauer, Mario Lucic, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv:1811.12359*, 2018.

SUPPLEMENTARY MATERIAL

A IDENTIFIABILITY THEORY WITH LATENT SPACES OF DIFFERENT DIMENSION

We need to prove that

$$\widetilde{\boldsymbol{T}}(\mathrm{z}) = \boldsymbol{A}\widetilde{\boldsymbol{T}}'(\mathrm{w}) + \boldsymbol{c} \tag{13}$$

where $\boldsymbol{A} \in \mathbb{R}^{nk \times n'k}$ is full rank (with rank nk) when the dimension n' of the estimating latent space is greater or equal in size to the dimension n of the generating latent space. First note that

$$\text{rank}(\boldsymbol{AB}) \leq \min(\text{rank}(\boldsymbol{A}), \text{rank}(\boldsymbol{B})) \tag{14}$$

for any matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. The proof is almost identical to that in Appendix B of Khemakhem et al. (2019), so only the points of difference will be noted.

All reasoning is identical (except for replacing n with n' on the right hand side of the equalities) until equation (24):

$$\boldsymbol{Q} = \boldsymbol{A}\boldsymbol{Q}' \tag{15}$$

where here $\boldsymbol{Q}' \in \mathbb{R}^{n'k \times nk}$. $\boldsymbol{Q}$ is invertible, so has rank nk, which means that $\boldsymbol{A}$ and $\boldsymbol{Q}'$ have rank at least nk, or else they would violate (14). The maximum rank $\boldsymbol{A}$ can have is nk, hence $\boldsymbol{A}$ has full rank.

B SPARSITY IN UNMIXING MATRIX: GAUSSIAN DISTRIBUTION

Suppose samples z from the generating latent space follow a conditional Gaussian distribution (equation (2)). Suppose the estimating model g faithfully reproduces the observed conditional density $p(\mathrm{x}|\mathrm{u})$ and samples w from its latent space also follow a conditional Gaussian distribution. Then we can apply equation (6) to relate the generating and estimating latent spaces.

Suppose arbitrary values of n and n', with $n' \geq n$. The sufficient statistics of a normal distribution with free mean and variance are z and z^2 (see equation (4)). Hence the relationship between the latent spaces becomes

$$\begin{pmatrix} \mathrm{z} \\ \mathrm{z}^2 \end{pmatrix} = \boldsymbol{A} \begin{pmatrix} \mathrm{w} \\ \mathrm{w}^2 \end{pmatrix} + \boldsymbol{c} \tag{16}$$

where the squaring is applied element-wise. We can write $\boldsymbol{A}$ in block matrix form as

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}^{(1)} & \boldsymbol{A}^{(2)} \\ \boldsymbol{A}^{(3)} & \boldsymbol{A}^{(4)} \end{pmatrix} \tag{17}$$

and $\boldsymbol{c}$ as

$$\boldsymbol{c} = \begin{pmatrix} \boldsymbol{c}^{(1)} \\ \boldsymbol{c}^{(2)} \end{pmatrix} \tag{18}$$

Then:

$$\mathrm{z} = \boldsymbol{A}^{(1)}\mathrm{w} + \boldsymbol{A}^{(2)}\mathrm{w}^2 + \boldsymbol{c}^{(1)} \tag{19}$$

$$\mathrm{z}^2 = \boldsymbol{A}^{(3)}\mathrm{w} + \boldsymbol{A}^{(4)}\mathrm{w}^2 + \boldsymbol{c}^{(2)} \tag{20}$$

so we can write for each dimension i of z

$$z_i = \sum_j A_{ij}^{(1)} w_j + \sum_j A_{ij}^{(2)} w_j^2 + c_i^{(1)} \tag{21}$$

$$z_i^2 = \sum_j A_{ij}^{(3)} w_j + \sum_j A_{ij}^{(4)} w_j^2 + c_i^{(2)} \tag{22}$$

In order to compare the equations, we need to square (21). To do so, we will have to square the second term on the right hand side, involving w_j^2. There is no matching term in (22), so we have to

set all entries of $\boldsymbol{A}^{(2)}$ to zero. In more detail:

$$\left(\sum_j A_{ij}^{(2)} w_j^2\right)^2 = \sum_j \sum_{j'} A_{ij}^{(2)} A_{ij'}^{(2)} w_j^2 w_{j'}^2 \tag{23}$$

$$= \sum_j (A_{ij}^{(2)})^2 w_j^4 + \sum_{j \neq j'} A_{ij}^{(2)} A_{ij'}^{(2)} w_j^2 w_{j'}^2 \tag{24}$$

The first term with w_j^4 matches no term in (22), so we have to set $A_{ij}^{(2)} = 0$ for all i and j. This simplifies the earlier equation:

$$z_i = \sum_j A_{ij}^{(1)} w_j + c_i^{(1)} \tag{25}$$

The square of the first term on the right hand side involves terms with $w_j w_{j'}$ cross terms:

$$\left(\sum_j A_{ij}^{(1)} w_j\right)^2 = \sum_j A_{ij}^{(1)} w_j^2 + \sum_{j \neq j'} A_{ij}^{(1)} A_{ij'}^{(1)} w_j w_{j'} \tag{26}$$

so we have to set $\boldsymbol{A}_{ij}^{(1)} \boldsymbol{A}_{ij'}^{(1)} = 0$ for all $j \neq j'$. This means that the i-th row of $\boldsymbol{A}^{(1)}$ can have at most one nonzero entry. It must also have at least one nonzero entry, since if the row were all zero, a row of $\boldsymbol{A}$ would be all zero (since $\boldsymbol{A}^{(2)} = 0$), but $\boldsymbol{A}$ has full rank. Since there are as many or fewer rows than columns ($n \leq n'$), each row of $\boldsymbol{A}$ is linearly independent, so it is not possible for one to be zero. Hence each row of $\boldsymbol{A}^{(1)}$ has exactly one nonzero entry. Moreover, no two rows of $\boldsymbol{A}^{(1)}$ have their nonzero entries in the same column. If they did, the two rows would not be linearly independent, but they must be since $\boldsymbol{A}$ has full rank. Therefore we can write

$$z_i = a_i w_j + b_i \tag{27}$$

where $a_i = A_{ij}^{(1)}$ and $b_i = c_i^{(1)}$. That is, the generating latent variable z_i is linearly related to some latent variable of the estimating model w_j. This estimated latent variable is uniquely associated with z_i and any estimated latent variables not associated with a generating latent variable z_i (in the case $n' > n$) encode no information about the generating latent space. So the model has decoded the original latent variables z up to an affine transformation and permutation as a subset of variables in its estimated latent space and has encoded no information (only noise) into the remaining latent variables.

C SPARSITY IN UNMIXING MATRIX: TWO-PARAMETER EXPONENTIAL FAMILY MEMBERS

The results of Appendix B can be extended to other members of the exponential family with 2 parameters. In the general case, writing $T_{i,1} = T_1$ and $T_{i,2} = T_2$ for all i, equations (21) and (22) become

$$T_1(z_i) = \sum_j A_{ij}^{(1)} T_1(w_j) + \sum_j A_{ij}^{(2)} T_2(w_j) + c_i^{(1)} \tag{28}$$

$$T_2(z_i) = \sum_j A_{ij}^{(3)} T_1(w_j) + \sum_j A_{ij}^{(4)} T_2(w_j) + c_i^{(2)} \tag{29}$$

which, with the definition $t = T_2 \circ T_1^{-1}$ becomes

$$T_1(z_i) = \sum_j A_{ij}^{(1)} T_1(w_j) + \sum_j A_{ij}^{(2)} t(T_1(w_j)) + c_i^{(1)} \tag{30}$$

$$t(T_1(w_j)) = \sum_j A_{ij}^{(3)} T_1(w_j) + \sum_j A_{ij}^{(4)} t(T_1(w_j)) + c_i^{(2)}. \tag{31}$$

We can combine these equations to get

$$t\left(\sum_j A_{ij}^{(1)} T_1(w_j) + \sum_j A_{ij}^{(2)} t(T_1(w_j)) + c_i^{(1)}\right) = \sum_j A_{ij}^{(3)} T_1(w_j) + \sum_j A_{ij}^{(4)} t(T_1(w_j)) + c_i^{(2)}. \tag{32}$$

This equation has many summed terms on the right. Suppose that t has a convergent Taylor expansion in some region of its domain. We can take this expansion of the term on the left and compare coefficients. Since t cannot be linear (otherwise the two sufficient statistics would be the same), there will be terms in the expansion of order two or higher. As in the Gaussian case, these polynomial terms create cross terms which are impossible to reconcile with those on the right hand side of the equation. The only consistent solutions can be found by setting all coefficients except those of functions of w_j (for some j) to zero:

$$t\left(A_{ij}^{(1)}T_1(w_j) + A_{ij}^{(2)}t(T_1(w_j)) + c_i^{(1)}\right) = A_{ij}^{(3)}T_1(w_j) + A_{ij}^{(4)}t(T_1(w_j)) + c_i^{(2)} \tag{33}$$

We can further simplify this equation by examining higher-order terms, but need to know the form of t to do so. In any case, we can now write equation (28) as

$$T_1(z_i) = A_{ij}^{(1)}T_1(w_j) + A_{ij}^{(2)}t(T_1(w_j)) + c_i^{(1)} \tag{34}$$

showing that each generating latent variable z_i is related to exactly one estimated latent variable w_j. As in the Gaussian case, we can use the full rank property of $\boldsymbol{A}$ to see that each estimated latent variable is associated with at most one generating latent variable, and any estimated latent variables not associated with any generating latent variables must encode only noise.

The task now is to check the form of t for each two-parameter member of the exponential family, to see what further constraints we can derive from equation (33). This will not be done in detail here, but the results are stated in Table 1.

Table 1: Two-parameter exponential family members and selected properties.

Distribution	Sufficient Statistics	$t(x)$	Latent Space Relationship
Normal	(z, z^2)	x^2	$z_i = aw_j + c$
Lognormal	$(\log z, (\log z)^2)$	x^2	$\log z_i = a \log w_j + c$
Inverse Gaussian	$(z, 1/z)$	$1/x$	$z_i = aw_j$ or $z_i = a/w_j$
Gamma	$(\log z, z)$	$\exp(x)$	$z_i = aw_j$
Inverse Gamma	$(\log z, 1/z)$	$\exp(-x)$	$z_i = aw_j$
Beta	$(\log z, \log(1-z))$	$\log(1-\exp(x))$	$\log z_i = \log w_j$ or $\log z_i = \log(1-w_j)$

D NETWORK ARCHITECTURE

The estimating model g is built in the reverse direction for practical purposes, so the models described here are g^{-1} which maps from the data space to the latent space. The type and number of coupling blocks for the different experiments are shown below. The affine coupling function is the concatentation of the scale function s and the translation function t, computed together for efficiency, as in Kingma & Dhariwal (2018). It is implemented as either a fully connected network (MLP) or convolutional network, with the specified layer widths and a ReLU activation after all but the final layers. For the convolutional coupling blocks, the splits are along the channel dimension. The scale function s is passed through a clamping function $2\tanh(s)$, which limits the output to the range (-2, 2), as in Ardizzone et al. (2019). Two affine coupling functions are applied per block, as described in Dinh et al. (2016). Downsampling increases the number of channels by a factor of 4 and decreases the image width and height by a factor of 2, done in a checkerboard-like manner, as described in Jacobsen et al. (2018). The dimensions are permuted in a random but fixed way before application of each fully connected coupling block and likewise the channels for the convolutional coupling blocks. The network for the artificial data experiments has 4,480 learnable parameters and the network for the EMNIST experiments has 2,620,192 learnable parameters.

Table 2: Network architecture for artificial data experiments

Type of block	Number	Input shape	Affine coupling function layer widths
Fully Connected Coupling	8	10	$5 \to 10 \to 10 \to 10$

Table 3: Network architecture for EMNIST experiments

Type of block	Number	Input shape	Affine coupling function layer widths
Downsampling	1	(1, 28, 28)	
Convolutional Coupling	4	(4, 14, 14)	$2 \to 16 \to 16 \to 4$
Downsampling	1	(4, 14, 14)	
Convolutional Coupling	4	(16, 7, 7)	$8 \to 32 \to 32 \to 16$
Flattening	1	(16, 7, 7)	
Fully Connected Coupling	2	784	$392 \to 392 \to 392 \to 784$

E FIGURES FROM THE ARTIFICIAL DATA EXPERIMENTS

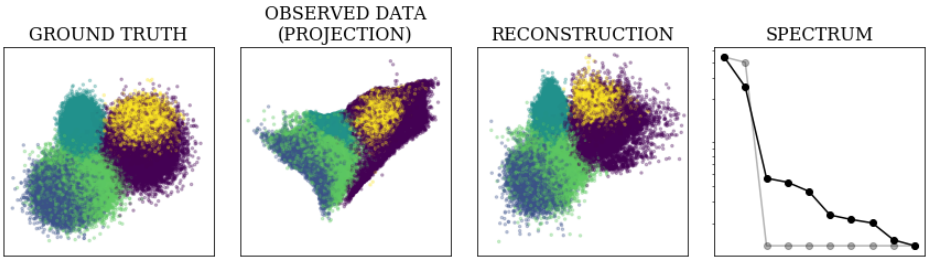


Figure 6: Experiment 1: Five mixture components and 100,000 data points. GIN successfully reconstructs the generating latent space and gives importance to only two of its ten latent variables, reflecting the two-dimensional nature of the generating latent space.

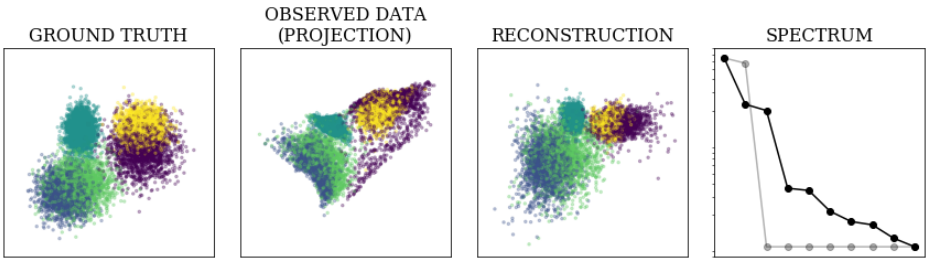
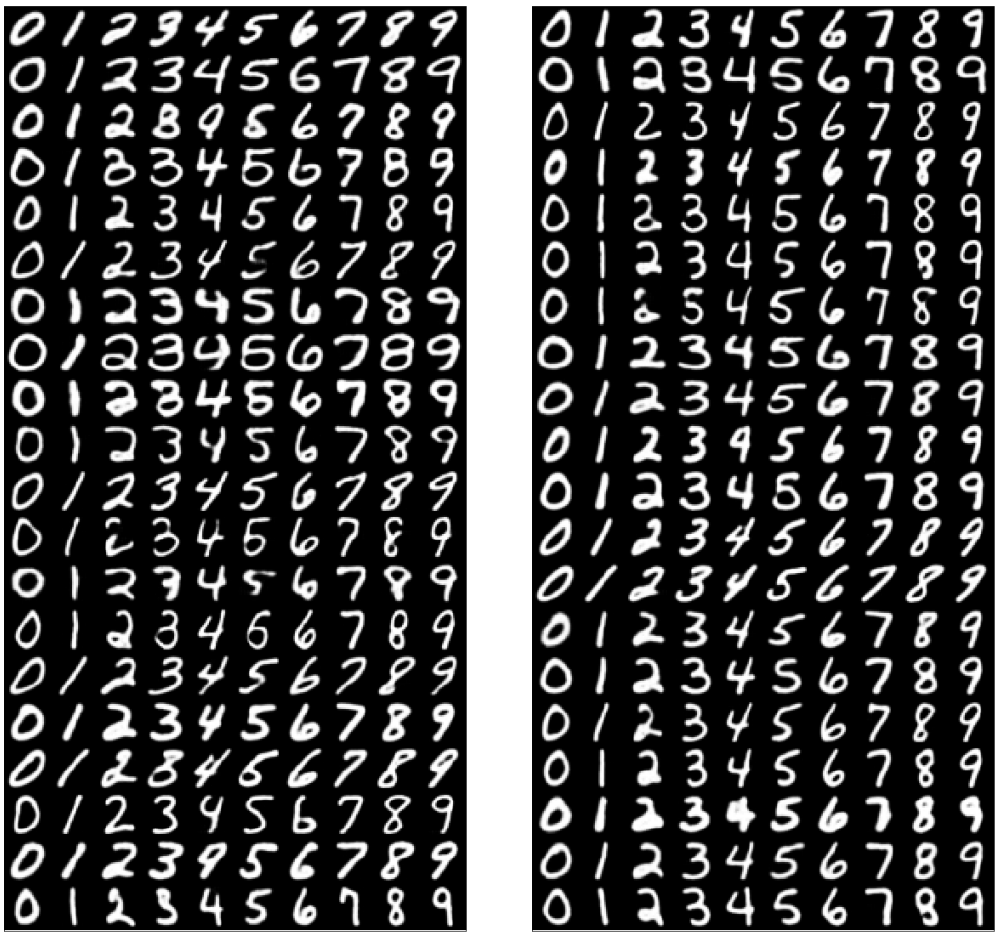


Figure 7: Same experiment as in Figure 6 with the number of data points reduced to 10,000. GIN fails to successfully estimate the ground truth latent variables, due to limited data: The first variable (x-axis) is well approximated in the reconstruction, but the second variable (y-axis) is split into two in the reconstruction, one capturing mainly information about the lower two clusters (shown here) and another information about the other mixture components (not shown). We also observe a less clear spectrum, where three variables are given more importance than the rest, not faithfully reflecting the two-dimensional nature of the generating latent space.

Figure 8: Experiment 2: Only three mixture components (not sufficient for identifiability according to the theory). Nevertheless, GIN successfully reconstructs the ground truth latent variables. This suggests that the current theory of nonlinear ICA relies on sufficient, but not necessary, conditions for identifiability.

F EMNIST FIGURES

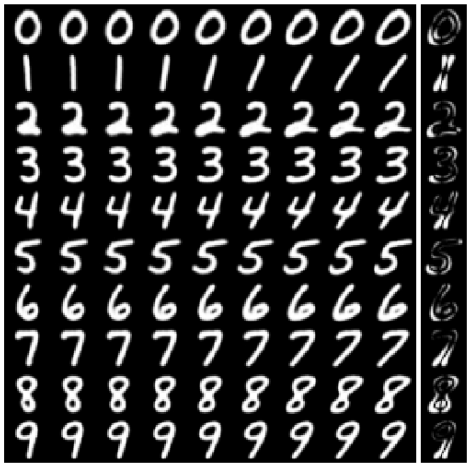


(a) Full Temperature

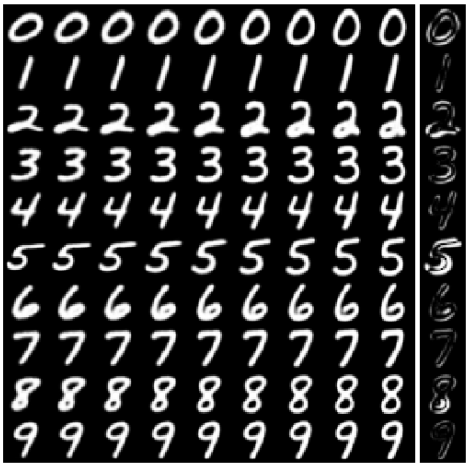
(b) Reduced Temperature (T = 0.8)

Figure 9: Full and reduced temperature samples from the model trained on EMNIST. Reduced temperature samples are made by sampling from a Gaussian distribution where the standard deviation is reduced by the temperature factor. The 22 most significant variables are sampled, with the others kept to their mean value. This eliminates noise from the images but preserves the full variability of digit shapes. Each row has the same latent code (whitened value) but is conditioned on a different class in each column, hence the style of the digits is consistent across rows.

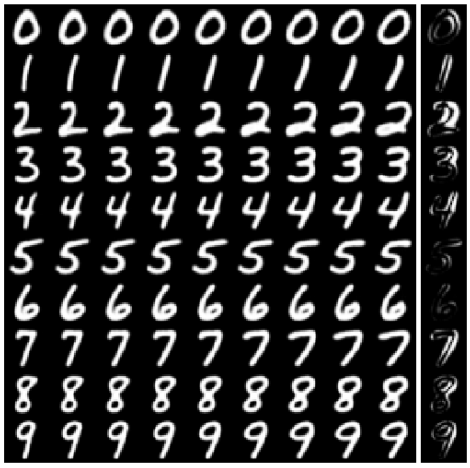
(a) Variable 1: width of top half



(b) Variable 2: slant/angle



(c) Variable 3: height

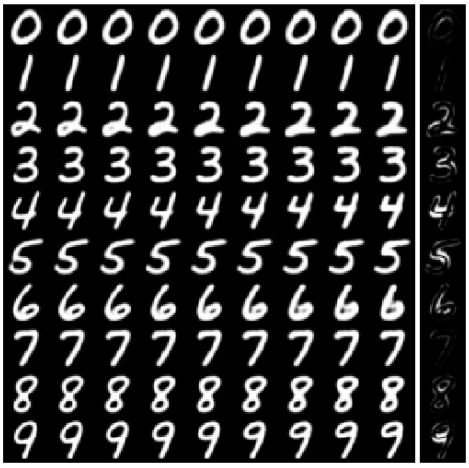


(d) Variable 4: bend through center

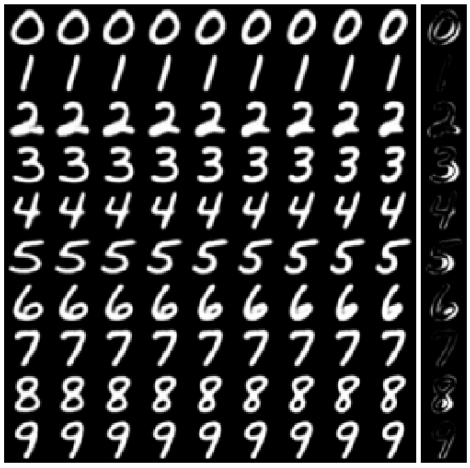
Figure 10: Most significant latent variables 1 to 4. Each row is conditioned on a different digit label. The variable runs from -2 to +2 standard deviations across the columns, with all other variables held constant at their mean value. The rightmost column shows a heatmap of the image areas most affected by the variable, computed as the absolute pixel difference between -1 and +1 standard deviations.

(a) Variable 5: line thickness

(b) Variable 6: slant of vertical bar in 4,7 and 9



(c) Variable 7: height of horizontal bar



(d) Variable 8: width of bottom half

Figure 11: Most significant latent variables 5 to 8. Each row is conditioned on a different digit label. The variable runs from -2 to +2 standard deviations across the columns, with all other variables held constant at their mean value. The rightmost column shows a heatmap of the image areas most affected by the variable, computed as the absolute pixel difference between -1 and +1 standard deviations.

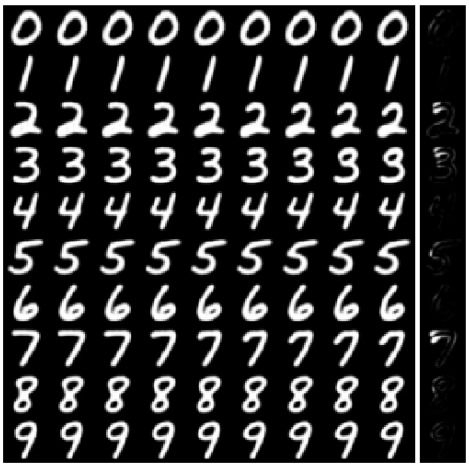
(a) Variable 9: extension of center left feature towards the left

(b) Variable 10: openness of lower loop

(c) Variable 11: shape of center right feature

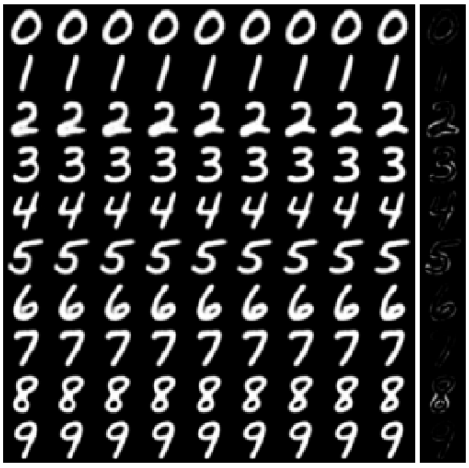
(d) Variable 12: top right corner

Figure 12: Most significant latent variables 9 to 12. Each row is conditioned on a different digit label. The variable runs from -2 to +2 standard deviations across the columns, with all other variables held constant at their mean value. The rightmost column shows a heatmap of the image areas most affected by the variable, computed as the absolute pixel difference between -1 and +1 standard deviations.

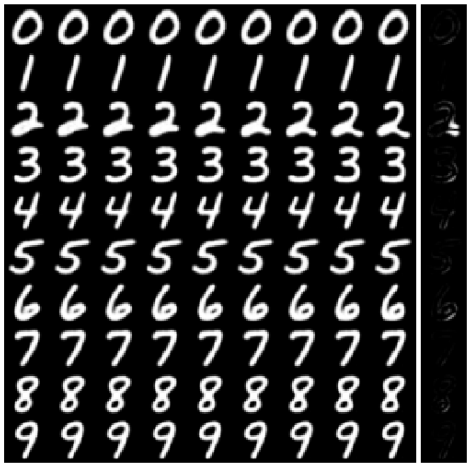


(a) Variable 13: orientation of stroke in top left corner for 2, 3 and 7

(b) Variable 14: shape of top part of bottom loop



(c) Variable 15: angle of centrally located horizontal features

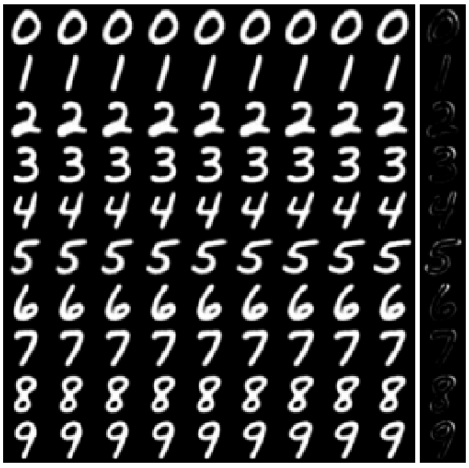


(d) Variable 16: bottom right stroke of 2

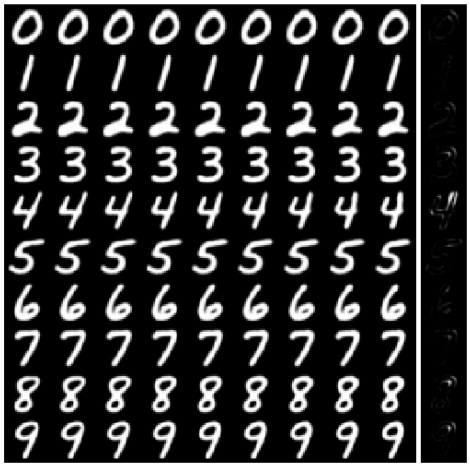
Figure 13: Most significant latent variables 13 to 16. Each row is conditioned on a different digit label. The variable runs from -2 to +2 standard deviations across the columns, with all other variables held constant at their mean value. The rightmost column shows a heatmap of the image areas most affected by the variable, computed as the absolute pixel difference between -1 and +1 standard deviations.

(a) Variable 17: top left stroke of 4

(b) Variable 18: top stroke of 5 and 7

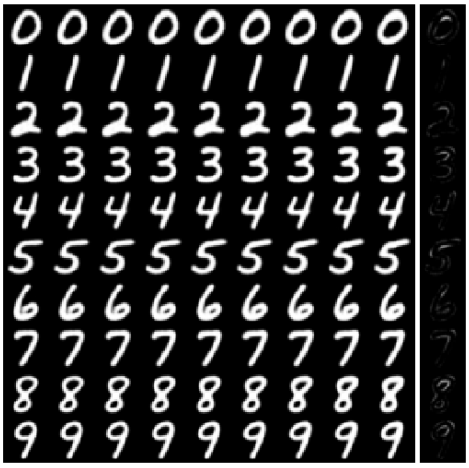


(c) Variable 19: extension towards top right



(d) Variable 20: curvature of vertical stroke of 4

Figure 14: Most significant latent variables 17 to 20. Each row is conditioned on a different digit label. The variable runs from -2 to +2 standard deviations across the columns, with all other variables held constant at their mean value. The rightmost column shows a heatmap of the image areas most affected by the variable, computed as the absolute pixel difference between -1 and +1 standard deviations.



(a) Variable 21: thickness of upper loop

(b) Variable 22: extension towards top left

(c) Variable 23: no effect

(d) Variable 24: no effect

Figure 15: Most significant latent variables 21 to 24. Each row is conditioned on a different digit label. The variable runs from -2 to +2 standard deviations across the columns, with all other variables held constant at their mean value. The rightmost column shows a heatmap of the image areas most affected by the variable, computed as the absolute pixel difference between -1 and +1 standard deviations.