IVAE-GAN

Identifiable VAE-GAN Models for Latent Representation Learning

Dideriksen, Bjorn Uttrup; Derosche, Kristoffer; Tan, Zheng Hua

Published in:
IEEE Access

DOI (link to publication from Publisher):
10.1109/ACCESS.2022.3172333

Creative Commons License
CC BY 4.0

Publication date:
2022

Document Version
Publisher's PDF, also known as Version of record

Link to publication from Aalborg University

Citation for published version (APA):
Dideriksen, B. U., Derosche, K., & Tan, Z. H. (2022). IVAE-GAN: Identifiable VAE-GAN Models for Latent Representation Learning. *IEEE Access*, *10*, 48405-48418. https://doi.org/10.1109/ACCESS.2022.3172333

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 21, 2022, accepted April 4, 2022, date of publication May 3, 2022, date of current version May 10, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3172333

iVAE-GAN: Identifiable VAE-GAN Models for Latent Representation Learning

BJØRN UTTRUP DIDERIKSEN, KRISTOFFER DEROSCHE, AND ZHENG-HUA TAN, (Senior Member, IEEE)
Department of Electronic Systems, Aalborg University, 9220 Aalborg, Denmark

Corresponding author: Bjørn Uttrup Dideriksen (bdider16@student.aau.dk)

This work was supported in part by Pioneer Centre for Artificial Intelligence, Denmark.

ABSTRACT Remarkable progress has been made within nonlinear Independent Component Analysis (ICA) and identifiable deep latent variable models. Formally, the latest nonlinear ICA theory enables us to recover the true latent variables up to a linear transformation by leveraging unsupervised deep learning. This is of significant importance for unsupervised learning in general as the true latent variables are of principal interest for meaningful representations. These theoretical results stand in stark contrast to the mostly heuristic approaches used for representation learning which do not provide analytical relations to the true latent variables. We extend the family of identifiable models by proposing an identifiable Variational Autoencoder (VAE) based Generative Adversarial Network (GAN) model we name iVAE-GAN. The latent space of most GANs, including VAE-GAN, is generally unrelated to the true latent variables. With iVAE-GAN we show the first principal approach to a theoretically meaningful latent space by means of adversarial training. We implement the novel iVAE-GAN architecture and show its identifiability, which is confirmed by experiments. The GAN objective is believed to be an important addition to identifiable models as it is one of the most powerful deep generative models. Furthermore, no requirements are imposed on the adversarial training leading to a very general model.

INDEX TERMS Identifiability, VAE-GAN, deep learning, latent representation learning.

I. INTRODUCTION

One of the biggest challenges facing machine learning and unsupervised learning in particular is meaningful representation learning. A very recent leap in representation learning is the understanding of identifiability in deep latent variable models by [1], [2]. *Identifiability* has origins in early econometrics as shown by the ''problem of confluent relations (or problem of arbitrary parameters)'' [3]. The formulation states that if two or more parametrizations of the same model lead to the same joint distribution over observed random variables they are indistinguishable on the basis of observations and therefore *unidentifiable*. Up until recently the general consensus in literature agreed that arbitrary nonlinear functions, such as those modeled by neural networks, were almost surely unidentifiable under the assumption of independent latent variables [4], [5]. It is now understood that identifiability results can be achieved if the model assumes *conditionally* independent latent variables. That is, given an additionally observed variable under which the latent variables are independent, they may be estimated up to a linear transformation and in certain cases reduced to a simple scaled permutation. The nonlinear maps from observable to latent variables need not preserve dimensionality, but if they do it is worth noting that the identifiability results become interpretable as nonlinear ICA [5].

It has been shown that identifiability can be achieved in Variational Autoencoders (VAE) [2], Energy-Based Models (EBM) [1] and General Incompressible-flow Networks (GIN) [6]. However, it has remained an open question whether identifiability is possible in Generative Adversarial Networks (GANs) [7].

Generative models are widely used and GANs are no exception. However, GANs do not learn a mapping from data space to latent space which, by definition, is necessary in order to construct an identifiable model. They do nonetheless rely on a latent space constructed from an explicit latent distribution from which samples are drawn as input to the generator (typically $\mathcal{N}(\mathbf{0}, \mathbf{I})$). In this work we leverage the new identifiability results to propose the first identifiable

The associate editor coordinating the review of this manuscript and approving it for publication was Li He.

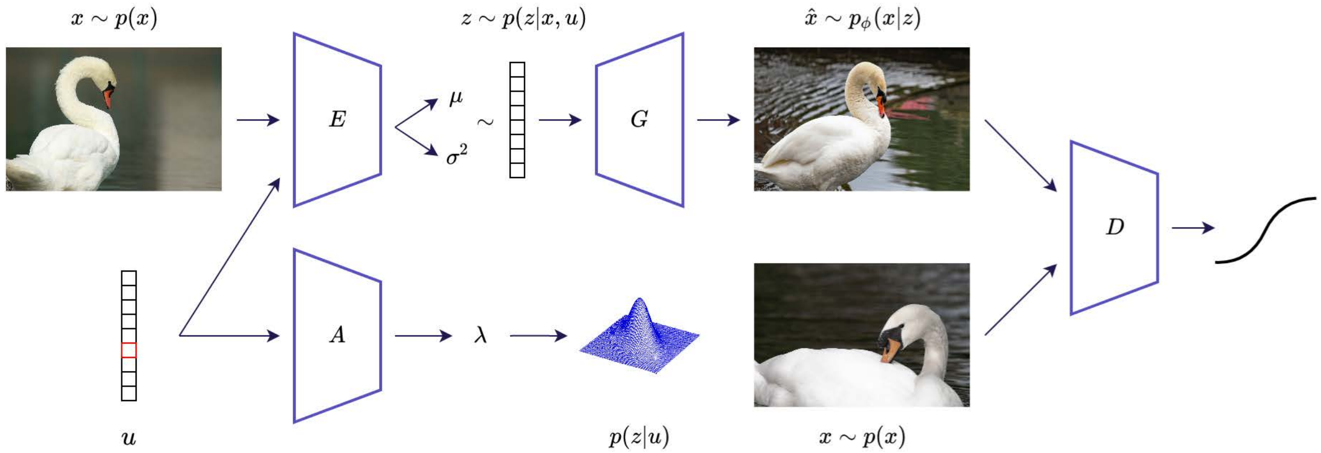

FIGURE 1. The proposed iVAE-GAN architecture. The latent variable model consists of four neural networks: an encoder *E* that learns the latent variables, a decoder/generator *G* that generates data in the original data space, a discriminator *D* that discriminates generated data from real data, and an auxiliary network *A* that learns the natural parameters, $\lambda_i(u)$, of an exponential family from the observed auxiliary variables.

GAN by using variational inference (iVAE-GAN) as shown in FIGURE 1.

It should be highlighted that in the proposed model, the latent distribution is not chosen but learnt through the encoder and auxiliary network and more importantly it is identifiable. This results in a model architecture that is similar to VAE-GAN [8], but with key differences as elaborated in the following. Most importantly iVAE-GAN is identifiable, thus requiring the additional auxiliary network as opposed to VAE-GAN. The generator of iVAE-GAN is updated according to the standard GAN loss and does not require a weighting of a VAE reconstruction loss and GAN loss. Thus iVAE-GAN is susceptible to improvements in the GAN minimax game while being applicable to the same problems with the same exceptional generative capabilities known from GAN literature. Due to identifiability the latent space of the GAN becomes meaningful as it is related to the true latent variables of the data. Therefore iVAE-GAN presents a principled approach to disentanglement in GAN that can be used to understand the true factors of variation in data. Thus, our model not only allows us to disentangle data into the true features but we can also use it to understand how the generative model recreates data and insert new latent code in the latent space to generate novel data.

Our contribution is three-fold:

1) First identifiability proof in GAN
2) An identifiable VAE-GAN model
3) We implement the novel iVAE-GAN model to validate and compare the identifiability of the proposed model against existing identifiable models.

II. RELATED WORK

This section reiterates the necessary theoretical results needed to propose iVAE-GAN. This is important, as it not only shows the theoretical justification for identifiability in iVAE-GAN, but also solidifies that the followed theoretical framework is general and could potentially be applied to a wider family of deep learning models.

A. IDENTIFIABILITY

A model is said to be identifiable if only one parametrization of the model can lead to the observed data distribution, i.e.

$$\textit{if } p_{\boldsymbol{\theta}_1}(\mathbf{x}) = p_{\boldsymbol{\theta}_2}(\mathbf{x}) \Rightarrow \boldsymbol{\theta}_1 = \boldsymbol{\theta}_2 \quad \forall(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) \tag{1}$$

On the other hand if $p_{\theta_1}(x) = p_{\theta_2}(x)$ but the parametrization is not unique, $\theta_1 \neq \theta_2$, the model is said to be unidentifiable on the basis of observations.

It is clear that identifiability is of interest in deep latent variable models as elaborated in the following. Latent variable models commonly model the joint distribution as:

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z}) \tag{2}$$

for $\mathbf{x} \in \mathbb{R}^d$, and $\mathbf{z} \in \mathbb{R}^n$ (lower-dimensional, $n \leq d$), but only provide training guarantees on the marginal distribution of the model (such as learning a lower bound on the observed marginal distribution):

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \tag{3}$$

Unfortunately deep latent variable model as in (2) do not learn the true joint distribution and as a result can not recover the original latent variables. In contrast, it is sufficient for identifiable models to learn the marginal distribution because only one parametrization of the joint distribution produces the seen marginal distribution and therefore the latent variables can be recovered.

B. IDENTIFIABLE MODEL

[2] and [1] have derived a very general deep latent variable model that is identifiable up to linear equivalence relations. Their work highlights that it is pivotal that the prior distribution is conditioned on an additional observed variable, u.

Therefore, the general form of the identifiable model becomes:

$$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}|\mathbf{u}) = p_{\mathbf{f}}(\mathbf{x}|\mathbf{z}) p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{u}) \tag{4}$$

where $\mathbf{u} \in \mathbb{R}^m$ is the auxiliary variable observed alongside the data and $\boldsymbol{\theta} = (\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda})$ are the parameters of the conditional generative model. The conditional latent distribution, $p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{u})$, is assumed to belong to an exponential family of independent variables:

$$p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{u}) = \prod_{i}^{n} \frac{Q_i(z_i)}{Z_i(\mathbf{u})} \exp\left[\sum_{j=1}^{k} T_{i,j}(z_i)\lambda_{i,j}(\mathbf{u})\right] \tag{5}$$

where $Q_i(z_i)$ is the base measure, $Z_i(u)$ is the normalization coefficient, $T_{i,j}(z_i)$ are the sufficient statistics and $\lambda_{i,j}(u)$ are the natural parameters of the family. The natural parameters of the distribution depending on $\mathbf{u}$ is learnt by the network we name A in our implementation. The assumption that the latent distribution must belong to an exponential family is not considered restrictive as it has been shown to have universal approximation capabilities by [9]. The decoder, $p_{\mathbf{f}}(\mathbf{x}|\mathbf{z})$, is defined as:

$$p_{\mathbf{f}}(\mathbf{x}|\mathbf{z}) = p_{\boldsymbol{\epsilon}}(\mathbf{x} - \mathbf{f}(\mathbf{z})) \tag{6}$$

allowing $\mathbf{x}$ to be decomposed into $\mathbf{x} = \mathbf{f}(\mathbf{z}) + \boldsymbol{\epsilon}$, where the noise is distributed according to $p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon})$ and $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^d$ is assumed injective.

C. AUXILIARY VARIABLE $\mathbf{u}$

The auxiliary variable $\mathbf{u}$ is pivotal to the definition of the identifiable model. The auxiliary variable is an observed variable such that the collected dataset contains pairwise observations of both the data, $\mathbf{x}$, and $\mathbf{u}$ such that $\mathcal{D} = \{(\mathbf{x}^{(1)}, \mathbf{u}^{(1)}), \dots, (\mathbf{x}^{(N)}, \mathbf{u}^{(N)})\}$. From (5) it can be seen that it is critical that the latent variables are independent given $\mathbf{u}$. It would be natural to wonder: How do we know the latent variables, which by definition are never observed, are independent given $\mathbf{u}$? In short; we do not know. Often this will be application specific and rely on knowledge of the data at hand. For most labeled datasets, such as MNIST, $\mathbf{u}$ could simply be the label.

D. IDENTIFIABILITY RESULT

The identifiability result states that, given the data we observe is the marginal distribution of some generating process with joint distribution conditioned on $\mathbf{u}$ with true generating parameters $(\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda})$:

$$p_{(\mathbf{f},\mathbf{T},\boldsymbol{\lambda})}(\mathbf{x}|\mathbf{u}) = \int p_{\mathbf{f}}(\mathbf{x}|\mathbf{z}) p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{u})\, \mathrm{d}\mathbf{z} \tag{7}$$

and a deep generative model of the same form learns to approximate the marginal distribution of observed data with parameters $(\tilde{\mathbf{f}}, \tilde{\mathbf{T}}, \tilde{\boldsymbol{\lambda}})$ such that:

$$p_{(\mathbf{f},\mathbf{T},\boldsymbol{\lambda})}(\mathbf{x}|\mathbf{u}) = p_{(\tilde{\mathbf{f}},\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}})}(\mathbf{x}|\mathbf{u}) = \int p_{\tilde{\mathbf{f}}}(\mathbf{x}|\mathbf{z}) p_{\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}}}(\mathbf{z}|\mathbf{u})\, \mathrm{d}\mathbf{z} \tag{8}$$

Then the parameters $(\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda})$ and $(\tilde{\mathbf{f}}, \tilde{\mathbf{T}}, \tilde{\boldsymbol{\lambda}})$ are said to be $\sim_A$ *−identifiable* such that:

$$(\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda}) \sim_A (\tilde{\mathbf{f}}, \tilde{\mathbf{T}}, \tilde{\boldsymbol{\lambda}}) \Leftrightarrow \mathbf{T}(\mathbf{f}^{-1}(\mathbf{x})) = A\tilde{\mathbf{T}}(\tilde{\mathbf{f}}^{-1}(\mathbf{x})) + \mathbf{c} \tag{9}$$

for some $nk \times nk$ invertible matrix A and vector $\mathbf{c}$. We provide a walk-through of the original proof in Appendix. Main points from the proof include that with a small assumption on the nature of the noise in (6) the underlying noiseless distributions of the models must be equal. By using said equality a system of equations can be constructed because the noiseless distributions are equal for all $\mathbf{u}$. This system of equations has a matrix representation of the form $L^T\mathbf{T}(\mathbf{f}^{-1}(\mathbf{x})) = \tilde{L}^T\tilde{\mathbf{T}}(\tilde{\mathbf{f}}^{-1}(\mathbf{x})) + \mathbf{b}$. The entries of the matrix L are a function of points of u. It is assumed that there exists at least $nk + 1$ points of u such that the matrix L is invertible. And lastly an assumption on the sufficient statistics, T, is made to prove the final equivalence relation.

This theoretical result is significant because it states that the trained deep generative model will have recovered the original latent variables, $\mathbf{f}^{-1}(\mathbf{x}) = \mathbf{z}$, up to a linear transformation of the sufficient statistics.

The theory requires that estimation models must follow a deep latent variable model with a conditional prior as seen from (4) and it must be able to approximate the seen marginal distribution. The proposed iVAE-GAN model learns both a variational approximation, $q_\phi(z|x, u)$, of the posterior, $p_\theta(z|u)$, and a generative model and is therefore an appropriate deep latent variable model. In the next section we show that the iVAE-GAN model also fulfills the second condition and thereby make the first link between identifiability and GAN.

III. iVAE-GAN

iVAE-GAN has a hybrid loss function that consists of a divergence loss for the encoding of the latent space with respect to the prior (conditional) distribution and an adversarial loss for the generated samples such that:

$$\mathcal{L}_{iVAE-GAN} = \mathcal{L}_{prior} + \mathcal{L}_{GAN} \tag{10}$$

where we define:

$$\mathcal{L}_{prior} = -KL(q_\phi(z|x, u)||p_\theta(z|u)) \tag{11}$$

and

$$\begin{aligned} \mathcal{L}_{GAN} &= V(D, G) \\ &= \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] \\ &\quad + \mathbb{E}_{z \sim p_z(z|u)}[\log(1 - D(G(z)))] \end{aligned} \tag{12}$$

Training is then performed according to:

$$\begin{aligned} \max_{\phi} \mathcal{L}_{prior} &+ \min_{G}\max_{D} \mathcal{L}_{GAN} \\ &= \max_{\phi} -KL(q_\phi(z|x, u)||p_\theta(z|u)) \\ &\quad + \min_{G}\max_{D} V(G, D) \end{aligned} \tag{13}$$

In the following we show that the loss is a lower bound on the difference between the log probability of the data and the expected log likelihood of the data generated by the decoder:

$$\log p(x) - \mathbb{E}_{z\sim q\phi(z|x,u)}[\log p_\Phi(x|z)] \geq \mathcal{L}_{iVAE-GAN} \tag{14}$$

and that by maximizing $\mathcal{L}_{iVAE-GAN}$ the expected log likelihood of the data generated by the decoder approaches the log probability of the data.

$\mathcal{L}_{prior}$ of the iVAE-GAN loss is related to the ELBO loss [10] such that:

$$\begin{aligned}&\log p(x) - \mathbb{E}_{z\sim q_\phi(z|x,u)}[\log p_\Phi(x|z)]\\&\quad\geq -KL(q_\phi(z|x,u)||p_\theta(z|u)) = \mathcal{L}_{prior}\end{aligned} \tag{15}$$

Now we show that the same inequality is also fulfilled by $\mathcal{L}_{prior} + \mathcal{L}_{GAN}$, but in contrast to (15) the data distribution may be learnt by maximizing $\mathcal{L}_{prior} + \mathcal{L}_{GAN}$. We assume an optimal discriminator, D^*, and use the result of [7]. Therefore we can write the optimization of $\mathcal{L}_{GAN}$ as: (See Appendix for proof).

$$\begin{aligned}\min_G \max_D \ \mathcal{L}_{GAN} &= \min_G V(G, D^*)\\&= \min_G \ -\log(4) + 2 \cdot JSD(p(x)||p_\Phi(x))\end{aligned} \tag{16}$$

To write our loss function only as a function that is to be maximized we pose the minimization over G as a maximization:

$$\begin{aligned}\min_G \ V(G, D^*) &\equiv \max_G \ -V(G, D^*)\\&= \max_G \ \log(4) - 2 \cdot JSD(p(x)||p_\Phi(x))\end{aligned} \tag{17}$$

Since we mean to maximize this function using a deep neural network the constant log(4) is inconsequential to the loss function. Therefore:

$$\min_G \ V(G, D^*) \equiv \max_G \ -2 \cdot JSD(p(x)||p_\Phi(x)) \tag{18}$$

Since the negated Jensen-Shannon divergence is non-positive it can always be added to the lesser side of an inequality without altering the inequality. Therefore we recover our lower bound by adding $-2 \cdot JSD(p(x)||p_\Phi(x|z))$ to (15):

$$\begin{aligned}&\log p(x) - \mathbb{E}_{z\sim q_\phi(z|x,u)}[\log p_\Phi(x|z)]\\&\quad\geq \underbrace{-KL(q_\phi(z|x,u)||p_\theta(z|u))}_{\mathcal{L}_{prior}} - \underbrace{2 \cdot JSD(p(x)||p_\Phi(x))}_{V(G,D^*)}\end{aligned} \tag{19}$$

The right-hand side of (19) can be recognized as the iVAE-GAN loss (updated according to (16) and (18)):

$$\mathcal{L}_{iVAE-GAN} = \mathcal{L}_{prior} - V(G, D^*) \tag{20}$$

Thus the iVAE-GAN loss is a lower bound on the difference between the log probability of observed data and expected log likelihood of the data generated by the decoder. To, hopefully, make (19) a little more interpretable we can make use of Jensen's inequality:

$$\mathbb{E}[\log(X)] \leq \log \mathbb{E}[X] \tag{21}$$

Therefore we can write:

$$\begin{aligned}&\log p(x) - \mathbb{E}_{z\sim q_\phi(z|x,u)}[\log p_\Phi(x|z)]\\&\quad\leq \log p(x) - \log \mathbb{E}_{z\sim q_\phi(z|x,u)}[p_\Phi(x|z)]\\&\quad= \log p(x) - \log \int p_\Phi(x|z)p_\phi(z)dz = \log p(x) - \log p_\Phi(x)\end{aligned} \tag{22}$$

By using the transitive property of inequalities we may write the lower bound as:

$$\begin{aligned}&\log p(x) - \log p_\Phi(x)\\&\qquad\geq -KL(q_\phi(z|x,u)||p_\theta(z|u))\\&\qquad\quad - 2 \cdot JSD(p(x)||p_\Phi(x))\end{aligned} \tag{23}$$

The Jensen-Shannon divergence measures the distance between two distributions and is therefore closely related to the difference of log probabilities, so as the lower bound is maximized the difference between log probabilities is minimized. In fact, the only condition for which the Jensen-Shannon divergence vanishes is $p(x) = p_\Phi(x)$, at which point the left-hand side becomes zero and the lower bound becomes:

$$0 \geq -KL(q_\phi(z|x,u)||p_\theta(z|u)) \tag{24}$$

Which is of course the normal bound for the negative KL divergence. Therefore, by maximizing $\mathcal{L}_{iVAE-GAN}$ we learn the data distribution while simultaneously maximizing the negative KL divergence between the encoded distribution, $q_\phi(z|x,u)$, and the prior distribution, $p_\theta(z|u)$ thus achieving an identifiable model.

This is, to the best of authors knowledge, the first work to actually extend and apply the theoretical framework to a deep latent variable model not contained in the original works by [1], [2]. The developed identifiability theory is claimed to be very general and extendable to a wide range of models and applications - a belief the authors of this paper share. Therefore it is not insignificant that we have shown how it extends to GAN. The training algorithm for iVAE-GAN is shown in Algorithm 1.

It is no coincidence that we consider GANs as a valuable framework to make identifiable. GANs are a very active area of research with state-of-the-art generative models and a wide range of applications. Providing proof of identifiability and initial experiments expand the toolkit researchers have at their disposal when meaningful latent spaces are desired in GANs. Importantly, the results we have shown do not impose or assume any restrictions on the adversarial training, therefore identifiability should be attainable in a large variety of GAN flavors that have other desirable properties such as stable training or alternative formulations of the minimax game.

IV. EXPERIMENTS

The core premise of the problem we aim to solve is that the latent variables are, by definition, never observed. Only the data which are a nonlinear function of the latent variables are

TABLE 1. Network sizes.

Model	Encoder	Decoder	Auxiliary	Discriminator
iVAE-GAN	$\{4, 4, 4\}$	$\{4, 4, 4\}$	$\{4, 4, 4\}$	$\left\{\begin{matrix} 32, & 32, & 32, & 32, & 32, & 32 \\ 32, & 32, & 32, & 32, & 32, & 32 \\ 256, & 256, & 256, & 256, & 256, & 256 \\ 1024, & 1024, & 1024, & 1024, & 1024, & 1024 \\ 1024, & 1024, & 1024, & 1024, & 1024, & 1024 \end{matrix}\right\}$
iVAE	$\{4, 4, 4\}$	$\{4, 4, 4\}$	$\{4, 4, 4\}$	-

Algorithm 1: iVAE-GAN Training Scheme

```
Initialize the model
Initialize the ADAM optimizer by [12]
while Training do
    for k discriminator steps do
        • Sample minibatch of $m$ samples from dataset $((x, u)^{(1)}, \ldots, (x, u)^{(m)})$
        • Encode $m$ samples into $m$ latent variable vectors $(z^{(1)}, \ldots, z^{(m)})$
        • Train model with ADAM
          $\mathcal{L}_{\mathcal{D}} = BinaryCrossEntropy(D(x), 1) + BinaryCrossEntropy(D(G(z), 0)$   (25)
    end
    for i generator steps do
        • Sample minibatch of $m$ samples from dataset $((x, u)^{(1)}, \ldots, (x, u)^{(m)})$
        • Encode $m$ samples into $m$ latent variable vectors $(z^{(1)}, \ldots, z^{(m)})$
        • Train model with ADAM
          $\mathcal{L} = BinaryCrossEntropy(D(G(z)), 1) - KL(q_{\phi}(z|x, u)||p_{\theta}(z|u))$   (26)
    end
end
```

observed. This premise is of great practical interest because it almost always reflects the true nature of data collection and the latent variables carry valuable information about the data. In our case where we also learn a generative model not only can we infer about the origin of the data but also generate unseen data.

However, this also means that datasets with known latent variables are very limited, even for datasets where the latent variables would intuitively be very simple. Consider e.g. MNIST. It would be very intuitive to expect the true latent space to consist of ten independent distributions - one for each number. Yet, because it is very difficult to observe the latent space we cannot use such datasets to validate our model. Therefore we have strictly used a synthetic dataset such that there is no ambiguity with respect to the true latent variables. Of course the latent variables are of greatest interest in real data but the scope of this work has been to show that identifiability is possible in adversarial networks.

A. DATASET

We have created our dataset with the data generator graciously provided in [1]. There are two main reasons for this choice: As discussed above datasets with known latent variables are very limited and secondly the same data generator has been used with other identifiable models, so it is a suitable generator to compare models across the same data and parameters. The data are generated in segments such that they become a non-stationary Gaussian time series. All segments are generated with equally many samples. The latent variables are drawn from an exponential family distribution with λ_i generated randomly and independent for each segment and passed through an uninitialized Multilayer Perceptron MLP to produce data that are a nonlinear function of the latent variables.

B. MCC METRIC

The Mean Correlation Coefficient (MCC) was used to quantify identifiability in [1], [2] and we adopt the same metric to quantify identifiability in iVAE-GAN. Given two sets of observations of m random variables each, the MCC metric calculates the interclass correlation coefficients (either Pearson or Spearman's correlation coefficients) between the m random variables of each set. Since every recovered latent variable should correspond to exactly one true latent variable a linear sum assignment problem is solved such that each recovered latent variable is assigned to exactly one true latent variable and the sum of the assigned correlation coefficients is maximized. The MCC score is then the mean of the correlation coefficients after assignment. A high MCC score thus reflects that the recovered latent variables are highly correlated with the true latent variables.

C. NETWORK ARCHITECTURE

TABLE 1 shows the size of the hidden dimensions of the networks, except for ICE-BeeM where we have left all parameters default (n_layers_flow = 10, ebm_hidden_size = 32). Hidden dimensions are fully connected MLPs with Leaky ReLU activation functions.

The reason multiple values are stated for the discriminator is because different widths were required for 100, 200, 500, 1000 and 2000 number of observations per segment. We believe this to be a result of increasing observations per segment which means the discriminator has to discriminate based on more points which in turn requires a larger network

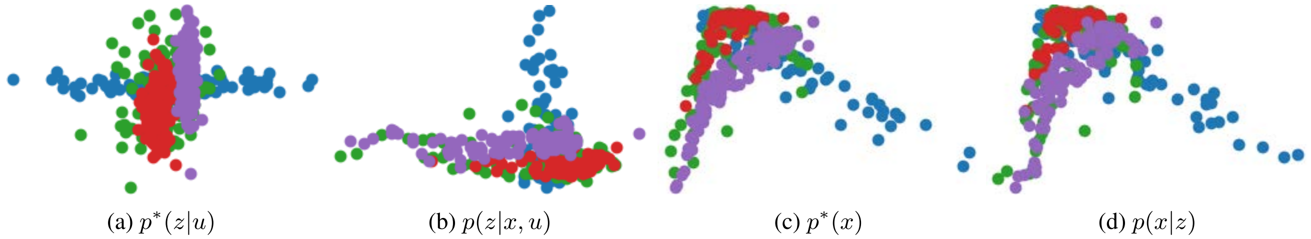

FIGURE 2. 2-Dimensional data and latent spaces. We have omitted axes to emphasize the linear indeterminacy as a rotation. a) The original generating latent variables. b) The recovered by iVAE-GAN. c) Input data. d) Data generated by iVAE-GAN.

in order to perform well. We also think this is consistent with common findings where more data points result in more complex GAN networks, e.g. it is harder to get reasonable results with higher resolution images than with lower resolution images.

D. OPTIMIZER

Two optimizers were used. One for the discriminator and one for the remaining three networks. The optimizers were both Adam optimizers with the same parameters: learning rate of 0.001 and $\beta = (0.5, 0.999)$.

E. TRAINING

All iVAE-GAN models were trained for 300000 iterations across 10 different seeds while iVAE and ICE-BeeM models were trained for 70000 iterations across 10 different seeds with the segment index as the auxiliary variable. iVAE used the default batch size of 256, ICE-BeeM also used the default batch size of 128. For this particular experiment of iVAE-GAN the batch size was simply set to the size of the dataset because the entire dataset could reside in the memory of the Tesla V100 32GB GPU made available by the university. For more details see Appendix.

During training we have noted the following remarks:

- iVAE-GAN inherits common training instabilities associated with adversarial training.
- Training is not consistently seen to monotonically converge. See Figure 6.
- Latent variables are only recovered well if the network learns to generate good data.
- Discriminator size is vital for well-behaved training. In practice we have used a VAE model to find a decoder network sufficiently complex to express the data and then tuned discriminator hyperparameters.

As it can be seen from FIGURE 2, iVAE-GAN generates similar but slightly different data, but most importantly it can be seen that the recovered latent variables are related to the true latent variables by a linear transformation, in this case a 90° clockwise rotation. To experimentally show identifiability we compare our model to iVAE and ICE-Beem [1], [2] as shown in FIGURE 3.

Our experiments indicate that identifiability is achievable not only in the model we have presented here, but in adversarial training generally without sacrificing the desired properties that make adversarial training appealing. Since the training stability of our model greatly resembles the notoriously challenging training of most GANs, future work could benefit from various developed methods aimed at stabilizing GAN training as described by [12]–[16].

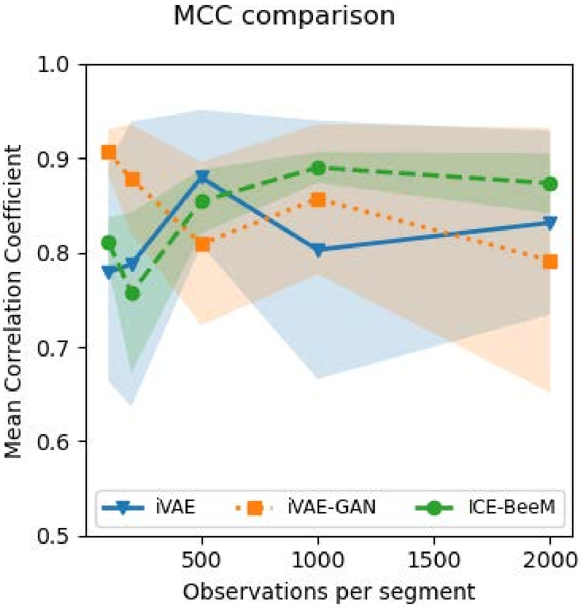

FIGURE 3. MCC comparison of iVAE, iVAE-GAN and ICE-BeeM with similar network parameters.

V. DISCUSSION

The main contribution of this paper is to show that GAN models can be made identifiable. This was achieved by showing that adversarial training fulfill the assumption that in the limit of infinite data the model learns the true data distribution while allowing the inference model to learn the prior conditional distribution, as shown in (24). In our comparison of iVAE-GAN (orange dotted) to the existing models iVAE (blue solid) and ICE-BeeM (green dashed) in FIGURE 3 we achieve comparable performance. In particular, iVAE-GAN performs better with few observations per segment than the other models but starts to struggle when more observations are available. We suspect this trend to be caused by our quite simple GAN implementation which tends to become unstable on large data. This could hopefully be remedied using various methods to stabilize training and regain performance when more observations per segment are used.

GAN has been associated with ethical concerns following the generation of images and speech that are virtually indistinguishable to the human eyes and ears. This work is not application specific and has therefore not produced any content that may be ethically concerning. Instead, we hypothesize that identifiability in GAN could be a remedying factor not because it prohibits malicious applications, but because it forces the generation to be based on the true generating process. Therefore there should be nothing inherently evil about an identifiable model. Identifiable models could in fact be of interest when sensitive or safety critical data are used because the operator runs little to no risk of introducing unwanted bias in the model.

VI. CONCLUSION

We have proven that Generative Adversarial Networks (GANs) can be made identifiable and proposed the identifiable model iVAE-GAN. As validation we have implemented the model and compared it to state-of-the-art identifiable models on the same data. This is the first proof of identifiability in GAN and it does not impose constraints on the adversarial training. Therefore, the results will apply broadly to a variety of different GAN flavors. We found through experiments, that the training dynamics of iVAE-GAN is prone to the same difficulties found in most GANs and is therefore a topic of interest for further work.

VII. FUTURE WORK

In future works the model should be tested on real data. This work has proven identifiability theoretically for iVAE-GAN as well as verified and compared its identifiability on a synthetic dataset. We propose that Amazon's Dinner Party Corpus (DiPCo) dataset would be ideal for this and other identifiable models as the cocktail party problem is the epitome of applied identifiability and DiPCo has unambiguous and easily interpretable latent variables.

APPENDIX. DERIVATION OF GLOBAL OPTIMALITY IN GAN

This derivation follows directly from [7] only stated more explicitly. GAN is formulated as a two-player minimax game according to:

$$\begin{aligned}\min_G \max_D V(G,D) &= \min_G \max_D \ \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] \\ &\quad + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \qquad (27)\end{aligned}$$

To simplify the derivations that are to follow, the second term is rewritten using the *law of the unconscious statistician* such that:

$$\begin{aligned}\min_G \max_D V(G,D) &= \min_G \max_D \ \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] \\ &\quad + \mathbb{E}_{x\sim p_g(x)}[\log(1-D(x))] \qquad (28)\end{aligned}$$

To express this in a way where the behaviour of the generator can be examined an optimal discriminator is assumed, such that the generator will try to minimize the following function:

$$\begin{aligned}C(G) &= \max_D V(G,D) \\ &= \max_D \int_x p_{data}(x)\log(D(x)) \\ &\quad + p_g(x)\log(1-D(x))\,\mathrm{d}x \qquad (29)\end{aligned}$$

This equation can be recognized as the function $f(y) = a\log(y) + b\log(1-y)$ which attains a maximum at $y = \frac{a}{a+b}$, which implies that the optimal discriminator is given by:

$$D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \qquad (30)$$

Thus, if $p_g = p_{data}$ the optimal discriminator will become $D^*_G(x) = \frac{1}{2}$ and by (28) we can find the optimum value at which the generated data will be indistinguishable from the true data:

$$\begin{aligned}V^*(G,D) &= \mathbb{E}_{x\sim p_{data}(x)}[\log(\frac{1}{2})] \\ &\quad + \mathbb{E}_{x\sim p_g(x)}[\log(1-\frac{1}{2})] \\ &= -\log 4 \qquad (31)\end{aligned}$$

Now we need to verify that $-\log 4$ is indeed a minimum of $V(G,D^*)$:

$$\begin{aligned}&\min_G V(G,D^*) \\ &= \min_G \int_x p_{data}(x)\log(\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}) \\ &\quad + p_g(x)\log(1-\frac{p_{data}(x)}{p_{data}(x)+p_g(x)})\,\mathrm{d}x \qquad (32)\end{aligned}$$

The first term can be recognized as a Kullback-Leibler divergence $KL(p_{data}||p_{data} + p_g) = \int_x p_{data}(x)\log(\frac{p_{data}(x)}{p_{data}(x)+p_g(x)})\,\mathrm{d}x$ and the second term can also be rewritten to a KL divergence since:

$$\begin{aligned}1 - \frac{p_{data}(x)}{p_{data}(x)+p_g(x)} &= \frac{p_{data}(x)+p_g(x)}{p_{data}(x)+p_g(x)} \\ - \frac{p_{data}(x)}{p_{data}(x)+p_g(x)} &= \frac{p_g(x)}{p_{data}(x)+p_g(x)} \qquad (33)\end{aligned}$$

Therefore (32) can be written as:

$$\begin{aligned}\min_G V(G,D^*) = \min_G \ &KL(p_{data}||p_{data}+p_g) \\ &+ KL(p_g||p_{data}+p_g) \qquad (34)\end{aligned}$$

The last step is achieved by rewriting the equation such that the two KL divergences can be expressed as a Jensen-Shannon divergence between p_{data} and p_g by multiplying the equation with $\frac{\log(2)}{\log(2)}$ and distributing terms:

$$\begin{aligned}&\min_G \ -\log(4) + KL(p_{data}||\frac{p_{data}+p_g}{2}) \\ &\quad + KL(p_g||\frac{p_{data}+p_g}{2}) \\ &= \min_G \ -\log(4) + 2\cdot JSD(p_{data}||p_g) \qquad (35)\end{aligned}$$

Since the Jensen-Shannon divergence is always non-negative and attains a minimum only when $p_{data} = p_g$, it is concluded that $\min_G V(G, D^*) = V^*(D, G) = -\log(4)$ only when $p_{data} = p_g$. In other words, the global optimum of adversarial training, under the assumption of an optimal discriminator, occurs only when the generated data follows the same distribution as that of the observed data.

APPENDIX. IDENTIFIABILITY PROOF

All the proofs stated herein comes from [2] only with a more explicit walk-through.

Given two sets of parameters $(\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda})$ and $(\tilde{\mathbf{f}}, \tilde{\mathbf{T}}, \tilde{\boldsymbol{\lambda}})$ such that $p_{\mathbf{f},\mathbf{T},\boldsymbol{\lambda}}(x|u) = p_{\tilde{\mathbf{f}},\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}}}(x|u)$ then the noise-free distributions, $\tilde{p}$ as they will be shown below, are also equal. The marginal distribution in the generative model can be written as:

$$\int_{\mathcal{Z}} p_{\mathbf{f}}(\mathbf{x}|\mathbf{z}) p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{u})\,\mathrm{d}\mathbf{z} = \int_{\mathcal{Z}} p_{\tilde{\mathbf{f}}}(\mathbf{x}|\mathbf{z}) p_{\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}}}(\mathbf{z}|\mathbf{u})\,\mathrm{d}\mathbf{z} \tag{36}$$

Substituting the decoder with the definition from (6) yields:

$$\begin{aligned}&\int_{\mathcal{Z}} p_{\boldsymbol{\epsilon}}(\mathbf{x} - \mathbf{f}(\mathbf{z})) p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{u})\,\mathrm{d}\mathbf{z} \\ &\quad = \int_{\mathcal{Z}} p_{\boldsymbol{\epsilon}}(\mathbf{x} - \tilde{\mathbf{f}}(\mathbf{z})) p_{\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}}}(\mathbf{z}|\mathbf{u})\,\mathrm{d}\mathbf{z}\end{aligned} \tag{37}$$

We now change the domain of the integral from $\mathcal{Z}$ to $\mathcal{X}$, by introducing $\bar{\mathbf{x}} = \mathbf{f}(\mathbf{z})$. We also introduce the notion of matrix volume denoted by $\mathrm{vol}\,A$, which acts as a replacement for the absolute determinant of the Jacobian introduced as a result of the change of variable:

$$\begin{aligned}&\int_{\mathcal{X}} p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{f}^{-1}(\bar{\mathbf{x}})|\mathbf{u}) \text{ vol } J_{\mathbf{f}^{-1}}(\bar{\mathbf{x}}) p_{\epsilon}(\mathbf{x} - \bar{\mathbf{x}})\,\mathrm{d}\bar{\mathbf{x}} \\ &\quad = \int_{\mathcal{X}} p_{\mathbf{T},\boldsymbol{\lambda}}(\tilde{\mathbf{f}}^{-1}(\bar{\mathbf{x}})|\mathbf{u}) \text{ vol } J_{\tilde{\mathbf{f}}^{-1}}(\bar{\mathbf{x}}) p_{\epsilon}(\mathbf{x} - \bar{\mathbf{x}})\,\mathrm{d}\bar{\mathbf{x}}\end{aligned} \tag{38}$$

We now introduce the following shorthand:

$$\tilde{p}_{\mathbf{T},\boldsymbol{\lambda},\mathbf{f},\mathbf{u}}(\mathbf{x}) = p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{f}^{-1}(\mathbf{x})|\mathbf{u}) \text{ vol } J_{\mathbf{f}^{-1}}(\mathbf{x}) \mathbb{1}_{\mathcal{X}}(\mathbf{x}) \tag{39}$$

where $\mathbb{1}_{\mathcal{X}}$ is the indicator function, assuring that the expression has measure zero if x is not contained in the image of $\mathbf{f}$:

$$\begin{aligned}&\int_{\mathbb{R}^d} \tilde{p}_{\mathbf{T},\boldsymbol{\lambda},\mathbf{f},\mathbf{u}}(\bar{\mathbf{x}}) p_{\epsilon}(\mathbf{x} - \bar{\mathbf{x}})\,\mathrm{d}\bar{\mathbf{x}} \\ &\quad = \int_{\mathbb{R}^d} \tilde{p}_{\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}},\tilde{\mathbf{f}},\mathbf{u}}(\bar{\tilde{x}}) p_{\epsilon}(\mathbf{x} - \bar{\mathbf{x}})\,\mathrm{d}\bar{\mathbf{x}}\end{aligned} \tag{40}$$

We recognize this to be the convolution between $\tilde{p}_{\mathbf{T},\boldsymbol{\lambda},\mathbf{f},\mathbf{u}}$ and p_{ϵ} as such:

$$(\tilde{p}_{\mathbf{T},\boldsymbol{\lambda},\mathbf{f},\mathbf{u}} * p_{\epsilon})(\mathbf{x}) = (\tilde{p}_{\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}},\tilde{\mathbf{f}},\mathbf{u}} * p_{\epsilon})(\mathbf{x}) \tag{41}$$

Transforming the functions to the Fourier domain allows us to simplify the expression further:

$$F[\tilde{p}_{\mathbf{T},\boldsymbol{\lambda},\mathbf{f},\mathbf{u}}](\omega)\phi_{\epsilon}(\omega) = F[\tilde{p}_{\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}},\tilde{\mathbf{f}},\mathbf{u}}](\omega)\phi_{\epsilon}(\omega) \tag{42}$$

Note here that we assume the characteristic function $\phi_{\epsilon}(\mathbf{x})$ to be non-zero for $\mathbf{x} \in \mathcal{X}$, which means it can be factored out yielding the final result, from which it is evident that:

$$F[\tilde{p}_{\mathbf{T},\boldsymbol{\lambda},\mathbf{f},\mathbf{u}}](\omega) = F[\tilde{p}_{\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}},\tilde{\mathbf{f}},\mathbf{u}}](\omega) \tag{43}$$

$$\tilde{p}_{\mathbf{T},\boldsymbol{\lambda},\mathbf{f},\mathbf{u}}(\mathbf{x}) = \tilde{p}_{\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}},\tilde{\mathbf{f}},\mathbf{u}}(\mathbf{x}) \tag{44}$$

Therefore the noise-free distributions has to be the same. In the following we wish to examine the relationship between the true parameters $(\mathbf{T}, \boldsymbol{\lambda}, \mathbf{f})$ and the estimated parameters $(\tilde{\mathbf{T}}, \tilde{\boldsymbol{\lambda}}, \tilde{\mathbf{f}})$ given that our model learns to accurately approximate the true data distribution $\tilde{p}_{\mathbf{T},\boldsymbol{\lambda},\mathbf{f},\mathbf{u}}(\mathbf{x})$. First we use (39) to write the expression for the marginal distribution:

$$\begin{aligned}\tilde{p}_{\mathbf{T},\boldsymbol{\lambda},\mathbf{f},\mathbf{u}}(\mathbf{x}) &= p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{f}^{-1}(\mathbf{x})|\mathbf{u}) \text{ vol } J_{\mathbf{f}^{-1}}(\mathbf{x}) \mathbb{1}_{\mathcal{X}}(\mathbf{x}) \\ &= p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{u}) \text{ vol } J_{\mathbf{f}^{-1}}(\mathbf{x}) \mathbb{1}_{\mathcal{X}}(\mathbf{x})\end{aligned} \tag{45}$$

Since $\mathbf{f}^{-1}(\mathbf{x}) = \mathbf{z}$ by definition. By inserting the expression for the prior distribution given an auxiliary variable, u: $p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{u}) = \prod_i^n \frac{Q_i(z_i)}{Z_i(\mathbf{u})} \exp\left[\sum_{j=1}^k T_{i,j}(z_i)\lambda_{i,j}(\mathbf{u})\right]$ from (5), we can write the marginal distribution over x as:

$$\begin{aligned}&\tilde{p}_{\mathbf{T},\boldsymbol{\lambda},\mathbf{f},\mathbf{u}}(\mathbf{x}) \\ &\quad = \prod_i^n \frac{Q_i(z_i)}{Z_i(\mathbf{u})} \exp\left[\sum_{j=1}^k T_{i,j}(z_i)\lambda_{i,j}(\mathbf{u})\right] \\ &\quad\quad \times \text{ vol } J_{\mathbf{f}^{-1}}(\mathbf{x}) \mathbb{1}_{\mathcal{X}}(\mathbf{x})\end{aligned} \tag{46}$$

Here we can safely drop the indicator function, $\mathbb{1}_{\mathcal{X}}(\mathbf{x})$, as the expression no longer contains integration and we will write $\mathbf{z} = \mathbf{f}^{-1}(\mathbf{x})$ again to emphasize that latent variables are inferred from data. For simplicity we shall work with the log pdf since it greatly simplifies the exponential term:

$$\begin{aligned}&\log \tilde{p}_{\mathbf{T},\boldsymbol{\lambda},\mathbf{f},\mathbf{u}}(\mathbf{x}) \\ &\quad = \sum_i^n (\log Q_i(\mathbf{f}_i^{-1}(\mathbf{x})) - \log Z_i(\mathbf{u}) \\ &\quad\quad + \sum_{j=1}^k (T_{i,j}(\mathbf{f}_i^{-1}(\mathbf{x}))\lambda_{i,j}(\mathbf{u}))) + \log \text{ vol } J_{\mathbf{f}^{-1}}(\mathbf{x})\end{aligned} \tag{47}$$

Thus we can rewrite (44) and investigate the relation between true and estimated parameters:

$$\begin{aligned}&\sum_i^n (\log Q_i(\mathbf{f}_i^{-1}(\mathbf{x})) - \log Z_i(\mathbf{u}) \\ &\quad + \sum_{j=1}^k (T_{i,j}(\mathbf{f}_i^{-1}(\mathbf{x}))\lambda_{i,j}(\mathbf{u}))) + \log \text{ vol } J_{\mathbf{f}^{-1}}(\mathbf{x}) \\ &= \sum_i^n (\log \tilde{Q}_i(\tilde{\mathbf{f}}_i^{-1}(\mathbf{x})) - \log \tilde{Z}_i(\mathbf{u}) \\ &\quad + \sum_{j=1}^k (\tilde{T}_{i,j}(\tilde{\mathbf{f}}_i^{-1}(\mathbf{x}))\tilde{\lambda}_{i,j}(\mathbf{u}))) + \log \text{ vol } J_{\tilde{\mathbf{f}}^{-1}}(\mathbf{x})\end{aligned} \tag{48}$$

Each side of the equation contain nk unknown parameters in $T_{i,j}$ and $\tilde{T}_{i,j}$ respectively, since they are summed over n

latent variables and k sufficient parameters per latent variable. Each side of the equation also has nk unknown parameters in $\lambda_{i,j}$ and $\tilde{\lambda}_{i,j}$. Therefore a system of equations is created for $nk+1$ different points $u^{(0)}, \ldots, u^{(nk)}$. In time series data divided into segments this step may intuitively be thought of as calculating the probability of a seeing a given sample in each of nk segments. It can also be seen as a consequence of (44) where we have equality between the marginal distribution for all choices of u. Thus we get the following system of equations:

$$\begin{aligned}\sum_i^n &(\log Q_i(\mathbf{f}_i^{-1}(\mathbf{x})) - \log Z_i(\mathbf{u_0}) \\ &\quad + \sum_{j=1}^k (T_{i,j}(\mathbf{f}_i^{-1}(\mathbf{x}))\lambda_{i,j}(\mathbf{u_0}))) + \log\ \text{vol}\ J_{\mathbf{f}^{-1}}(\mathbf{x}) \\ &= \sum_i^n (\log \tilde{Q}_i(\tilde{\mathbf{f}}_i^{-1}(\mathbf{x})) - \log \tilde{Z}_i(\mathbf{u_0}) \\ &\quad + \sum_{j=1}^k (\tilde{T}_{i,j}(\tilde{\mathbf{f}}_i^{-1}(\mathbf{x}))\tilde{\lambda}_{i,j}(\mathbf{u_0}))) + \log\ \text{vol}\ J_{\tilde{\mathbf{f}}^{-1}}(\mathbf{x}) \end{aligned} \tag{49}$$

$$\begin{aligned}\sum_i^n &(\log Q_i(\mathbf{f}_i^{-1}(\mathbf{x})) - \log Z_i(\mathbf{u_1}) \\ &\quad + \sum_{j=1}^k (T_{i,j}(\mathbf{f}_i^{-1}(\mathbf{x}))\lambda_{i,j}(\mathbf{u_1}))) + \log\ \text{vol}\ J_{\mathbf{f}^{-1}}(\mathbf{x}) \\ &= \sum_i^n (\log \tilde{Q}_i(\tilde{\mathbf{f}}_i^{-1}(\mathbf{x})) - \log \tilde{Z}_i(\mathbf{u_1}) \\ &\quad + \sum_{j=1}^k (\tilde{T}_{i,j}(\tilde{\mathbf{f}}_i^{-1}(\mathbf{x}))\tilde{\lambda}_{i,j}(\mathbf{u_1}))) + \log\ \text{vol}\ J_{\tilde{\mathbf{f}}^{-1}}(\mathbf{x}) \end{aligned} \tag{50}$$

$$\vdots$$

$$\begin{aligned}\sum_i^n &(\log Q_i(\mathbf{f}_i^{-1}(\mathbf{x})) - \log Z_i(\mathbf{u_{nk}}) \\ &\quad + \sum_{j=1}^k (T_{i,j}(\mathbf{f}_i^{-1}(\mathbf{x}))\lambda_{i,j}(\mathbf{u_{nk}}))) + \log\ \text{vol}\ J_{\mathbf{f}^{-1}}(\mathbf{x}) \\ &= \sum_i^n (\log \tilde{Q}_i(\tilde{\mathbf{f}}_i^{-1}(\mathbf{x})) - \log \tilde{Z}_i(\mathbf{u_{nk}}) \\ &\quad + \sum_{j=1}^k (\tilde{T}_{i,j}(\tilde{\mathbf{f}}_i^{-1}(\mathbf{x}))\tilde{\lambda}_{i,j}(\mathbf{u}_{nk}))) + \log\ \text{vol}\ J_{\tilde{\mathbf{f}}^{-1}}(\mathbf{x}) \end{aligned} \tag{51}$$

A neat trick is used to simplify this system of equations by using any of the $nk+1$ equations as pivot, we shall simply use $\mathbf{u_0}$. By pivot it is understood that we consider a ratio of pdfs or in this case, since we are dealing with logarithms, a difference of log pdfs. This is the motivation for using $nk+1$ points in our system of equations as we use one equation to pivot such that we end up with a system of nk equations. The consequence of this choice is that our equations no longer express how likely a sample is with a given $\mathbf{u_l}$ but rather how likely it is compared to $\mathbf{u_0}$, but this is of little importance as we are interested in the relation between parameters of the models and not the exact likelihood of seen samples. Therefore the system of equations becomes:

$$0 = 0 \tag{52}$$

$$\begin{aligned}\sum_i^n &(\log Q_i(\mathbf{f}_i^{-1}(\mathbf{x})) - \log Z_i(\mathbf{u_1}) \\ &\quad + \sum_{j=1}^k (T_{i,j}(\mathbf{f}_i^{-1}(\mathbf{x}))\lambda_{i,j}(\mathbf{u_1}))) + \log\ \text{vol}\ J_{\mathbf{f}^{-1}}(\mathbf{x}) \\ &\quad - (\sum_i^n (\log Q_i(\mathbf{f}_i^{-1}(\mathbf{x})) - \log Z_i(\mathbf{u_0}) \\ &\quad + \sum_{j=1}^k (T_{i,j}(\mathbf{f}_i^{-1}(\mathbf{x}))\lambda_{i,j}(\mathbf{u_0}))) + \log\ \text{vol}\ J_{\mathbf{f}^{-1}}(\mathbf{x})) \\ &= \sum_i^n (\log \tilde{Q}_i(\tilde{\mathbf{f}}_i^{-1}(\mathbf{x})) - \log \tilde{Z}_i(\mathbf{u_1}) \\ &\quad + \sum_{j=1}^k (\tilde{T}_{i,j}(\tilde{\mathbf{f}}_i^{-1}(\mathbf{x}))\tilde{\lambda}_{i,j}(\mathbf{u_1}))) + \log\ \text{vol}\ J_{\tilde{\mathbf{f}}^{-1}}(\mathbf{x}) \\ &\quad - (\sum_i^n (\log \tilde{Q}_i(\tilde{\mathbf{f}}_i^{-1}(\mathbf{x})) - \log \tilde{Z}_i(\mathbf{u_0}) \\ &\quad + \sum_{j=1}^k (\tilde{T}_{i,j}(\tilde{\mathbf{f}}_i^{-1}(\mathbf{x}))\tilde{\lambda}_{i,j}(\mathbf{u_0}))) + \log\ \text{vol}\ J_{\tilde{\mathbf{f}}^{-1}}(\mathbf{x})) \end{aligned} \tag{53}$$

$$\vdots$$

$$\begin{aligned}\sum_i^n &(\log Q_i(\mathbf{f}_i^{-1}(\mathbf{x})) - \log Z_i(\mathbf{u_{nk}}) \\ &\quad + \sum_{j=1}^k (T_{i,j}(\mathbf{f}_i^{-1}(\mathbf{x}))\lambda_{i,j}(\mathbf{u_{nk}}))) + \log\ \text{vol}\ J_{\mathbf{f}^{-1}}(\mathbf{x}) \\ &\quad - (\sum_i^n (\log Q_i(\mathbf{f}_i^{-1}(\mathbf{x})) - \log Z_i(\mathbf{u_0}) \\ &\quad + \sum_{j=1}^k (T_{i,j}(\mathbf{f}_i^{-1}(\mathbf{x}))\lambda_{i,j}(\mathbf{u_0}))) + \log\ \text{vol}\ J_{\mathbf{f}^{-1}}(\mathbf{x})) \\ &= \sum_i^n (\log \tilde{Q}_i(\tilde{\mathbf{f}}_i^{-1}(\mathbf{x})) - \log \tilde{Z}_i(\mathbf{u_{nk}}) \\ &\quad + \sum_{j=1}^k (\tilde{T}_{i,j}(\tilde{\mathbf{f}}_i^{-1}(\mathbf{x}))\tilde{\lambda}_{i,j}(\mathbf{u}_{nk}))) + \log\ \text{vol}\ J_{\tilde{\mathbf{f}}^{-1}}(\mathbf{x}) \\ &\quad - (\sum_i^n (\log \tilde{Q}_i(\tilde{\mathbf{f}}_i^{-1}(\mathbf{x})) - \log \tilde{Z}_i(\mathbf{u_0}) \end{aligned}$$

$$+\sum_{j=1}^{k}(\tilde{T}_{i,j}(\tilde{\mathbf{f}}_i^{-1}(\mathbf{x}))\tilde{\lambda}_{i,j}(\mathbf{u_0}))) + \log\ \mathrm{vol}\, J_{\tilde{\mathbf{f}}^{-1}}(\mathbf{x})) \tag{54}$$

By eliminating terms we get rid of $\log \tilde{Q}_i(\tilde{\mathbf{f}}_i^{-1}(\mathbf{x})$ and interestingly $\log\ \mathrm{vol}\, J_{\tilde{\mathbf{f}}^{-1}}(\mathbf{x})$ which is typically notoriously difficult to evaluate, therefore these equations can be reduced to:

$$0 = 0 \tag{55}$$

$$\sum_{i}^{n}(-\log Z_i(\mathbf{u_1}) + \sum_{j=1}^{k}(T_{i,j}(\mathbf{f}_i^{-1}(\mathbf{x}))\lambda_{i,j}(\mathbf{u_1})))$$
$$-(\sum_{i}^{n}(-\log Z_i(\mathbf{u_0}) + \sum_{j=1}^{k}(T_{i,j}(\mathbf{f}_i^{-1}(\mathbf{x}))\lambda_{i,j}(\mathbf{u_0})))$$
$$= \sum_{i}^{n}(-\log \tilde{Z}_i(\mathbf{u_1}) + \sum_{j=1}^{k}(\tilde{T}_{i,j}(\tilde{\mathbf{f}}_i^{-1}(\mathbf{x}))\tilde{\lambda}_{i,j}(\mathbf{u_1})))$$
$$-(\sum_{i}^{n}(-\log \tilde{Z}_i(\mathbf{u_0}) + \sum_{j=1}^{k}(\tilde{T}_{i,j}(\tilde{\mathbf{f}}_i^{-1}(\mathbf{x}))\tilde{\lambda}_{i,j}(\mathbf{u_0}))) \tag{56}$$

$$\vdots$$

$$\sum_{i}^{n}(-\log Z_i(\mathbf{u_{nk}}) + \sum_{j=1}^{k}(T_{i,j}(\mathbf{f}_i^{-1}(\mathbf{x}))\lambda_{i,j}(\mathbf{u_{nk}})))$$
$$-(\sum_{i}^{n}(-\log Z_i(\mathbf{u_0}) + \sum_{j=1}^{k}(T_{i,j}(\mathbf{f}_i^{-1}(\mathbf{x}))\lambda_{i,j}(\mathbf{u_0})))$$
$$= \sum_{i}^{n}(-\log \tilde{Z}_i(\mathbf{u_{nk}}) + \sum_{j=1}^{k}(\tilde{T}_{i,j}(\tilde{\mathbf{f}}_i^{-1}(\mathbf{x}))\tilde{\lambda}_{i,j}(\mathbf{u}_{nk})))$$
$$-(\sum_{i}^{n}(-\log \tilde{Z}_i(\mathbf{u_0}) + \sum_{j=1}^{k}(\tilde{T}_{i,j}(\tilde{\mathbf{f}}_i^{-1}(\mathbf{x}))\tilde{\lambda}_{i,j}(\mathbf{u_0}))) \tag{57}$$

By factoring terms and distributing sums for an arbitrary point $u^{(l)}$ we get:

$$\sum_{i}^{n}(\sum_{j=1}^{k}(T_{i,j}(\mathbf{f}_i^{-1}(\mathbf{x}))(\lambda_{i,j}(\mathbf{u_l}) - \lambda_{i,j}(\mathbf{u_0})))$$
$$+\sum_{i}^{n}\log\frac{Z_i(\mathbf{u_0})}{Z_i(\mathbf{u_l})}$$
$$= \sum_{i}^{n}(\sum_{j=1}^{k}(\tilde{T}_{i,j}(\tilde{\mathbf{f}}_i^{-1}(\mathbf{x}))(\tilde{\lambda}_{i,j}(\mathbf{u_l}) - \tilde{\lambda}_{i,j}(\mathbf{u_0})))$$
$$+\sum_{i}^{n}\log\frac{\tilde{Z}_i(\mathbf{u_0})}{\tilde{Z}_i(\mathbf{u_l})} \tag{58}$$

$T_{i,j}$ and $\lambda_{i,j}$ are elements from the tall vectors $\mathbf{T}$ and $\boldsymbol{\lambda}$ therefore the first term can be recognized as the inner product between $\mathbf{T}(\mathbf{f^{-1}}(\mathbf{x}))$ and $\bar{\boldsymbol{\lambda}}(\mathbf{u_l})$ where $\bar{\boldsymbol{\lambda}}(\mathbf{u_l})$ is defined as:

$$\bar{\boldsymbol{\lambda}}(\mathbf{u_l}) = \boldsymbol{\lambda}(\mathbf{u_l}) - \boldsymbol{\lambda}(\mathbf{u_0}) \tag{59}$$

Therefore (58) can be written as:

$$\left\langle \mathbf{T}(\mathbf{f^{-1}}(\mathbf{x})), \bar{\boldsymbol{\lambda}}(\mathbf{u_l}) \right\rangle = \left\langle \tilde{\mathbf{T}}(\tilde{\mathbf{f}}^{-1}(\mathbf{x})), \bar{\tilde{\boldsymbol{\lambda}}}(\mathbf{u_l}) \right\rangle + b_l$$
$$where\ \ b_l = \sum_{i}^{n}\log\frac{\tilde{Z}_i(\mathbf{u_0})Z_i(\mathbf{u_l})}{\tilde{Z}_i(\mathbf{u_l})Z_i(\mathbf{u_0})} \tag{60}$$

Across all nk equations $\mathbf{T}(\mathbf{f^{-1}}(\mathbf{x}))$ will be the same, therefore we can collect all the equations in a single matrix product by defining the $nk \times nk$ matrix $L = \left[\bar{\boldsymbol{\lambda}}(\mathbf{u_1}), \bar{\boldsymbol{\lambda}}(\mathbf{u_2}) \ldots \bar{\boldsymbol{\lambda}}(\mathbf{u_{nk}})\right]$ and $\mathbf{b} = \left[b_1, b_2, \ldots, b_{nk}\right]$such that:

$$L^T\mathbf{T}(\mathbf{f^{-1}}(\mathbf{x})) = \tilde{L}^T\tilde{\mathbf{T}}(\tilde{\mathbf{f}}^{-1}(\mathbf{x})) + \mathbf{b} \tag{61}$$

In the final step we assume that the true matrix of natural parameters, $\mathbf{L}$, is invertible to obtain the following result:

$$\mathbf{T}(\mathbf{f^{-1}}(\mathbf{x})) = A\tilde{\mathbf{T}}(\tilde{\mathbf{f}}^{-1}(\mathbf{x})) + \mathbf{c} \tag{62}$$

where $A = L^{T^{-1}}\tilde{L}^T$ and $\mathbf{c} = L^{T^{-1}}\mathbf{b}$. Thus, we see that the true latent variables are linear transformation of the recovered latent variables. The last step is to prove an equivalence relation such that the opposite is also true. That the recovered latents are also a linear transformation of the true latent variables. To do so it is assumed that the Jacobian of $\mathbf{T}$ exists has full rank n. Therefore:

$$J_{\mathbf{T}}(x) = AJ_{\tilde{\mathbf{T}}\circ\tilde{\mathbf{f}}^{-1}}(x) \tag{63}$$

By using the following inequality for the rank of a matrix multiplication we may deduce that the rank of both A and $J_{\tilde{\mathbf{T}}\circ\tilde{\mathbf{f}}^{-1}}$ is at least n:

$$Rank(AB) \le min(Rank(A), Rank(B)) \tag{64}$$

Since $J_{\tilde{\mathbf{T}}\circ\tilde{\mathbf{f}}^{-1}}$ is a $nk \times n$ matrix we can conclude that it exists and has full rank. And if $k = 1$ then A will be a square $n \times n$ matrix with full rank and thus invertible, such that (62) can be shown to be true in both directions.

For $k > 1$ the matrix A must be invertible in order to establish the equivalence relation. In the following we show how A is invertible under the assumption that each latent variable follows a *strongly exponential distribution*. A strongly exponential distribution is one that almost certainly contains the exponent and thus can not be reduced to the base measure. Formally:

$$(\exists\boldsymbol{\theta} \in \mathbb{R}^k \mid \forall x \in \mathcal{X}, \left\langle \mathbf{T}(\mathbf{x}), \boldsymbol{\theta} \right\rangle = const)$$
$$\Rightarrow (l(\mathcal{X}) = 0\ \ or\ \ \boldsymbol{\theta} = \mathbf{0}) \tag{65}$$

Which means that the exponent of a strongly exponential distribution only reduces to a constant if $\boldsymbol{\theta} = 0$ which means the inner product becomes zero, $\left\langle \mathbf{T}(\mathbf{x}), \mathbf{0} \right\rangle = 0$, or if the set $\mathcal{X}$ has Lebesgue measure 0. The following three Lemmas is used to derive useful properties for the derivate of the sufficient statistic, $\mathbf{T}'(x)$, from a strongly exponential distribution that is of relevance for the Jacobian matrix. The dimension, k, of all considered distributions is assumed minimal. That is, the distributions can not be rewritten with a $k' < k$.

A. LEMMA 1

Consider an exponential family distribution with $k \geq 2$ components. [...], the components of the sufficient statistic $\mathbf{T}$ are linearly independent.

If the components of $\mathbf{T}$ are not linearly independent then one of the components, $T_k(x)$, could be written as a combination of the remaining components for an $\mathbf{a} \neq \mathbf{0}$.

$$T_k(x) = \sum_{i}^{k-1} a_i T_i(x) \tag{66}$$

If that was possible, we would have contradicted the assumption that the dimension of the distribution, k, is minimal.

B. LEMMA 2

Consider a strongly exponential family distribution such that its sufficient statistic T is differentiable almost surely. Then $T_i' \neq 0$ almost everywhere on $\mathbb{R}$ for all $1 \leq \mathrm{i} \leq \mathrm{k}$

We provide an alternate proof than the original, simply because we used this alternate proof to verify our understanding of the original proof. If we consider an exponential distribution that is *not* strongly exponential then we necessarily have:

$$\langle \mathbf{T}(\mathbf{x}), \boldsymbol{\theta} \rangle = T_1(x)\theta_1 + T_2(x)\theta_1 + \cdots + T_k\theta_k = const \tag{67}$$

The derivative would then become:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}x} \langle \mathbf{T}(\mathbf{x}), \boldsymbol{\theta} \rangle &= \langle \mathbf{T}(\mathbf{x})', \boldsymbol{\theta} \rangle \\ &= T_1'(x)\theta_1 + T_2'(x)\theta_2 + \cdots + T_k'(x)\theta_k \\ &= 0 \end{aligned} \tag{68}$$

Thus for an exponential distribution that is not strongly exponential the derivative of the exponent must be equal to zero. Which can be achieved in many different ways having either $\boldsymbol{\theta} = \mathbf{0}$, $\mathbf{T}(x) = \mathbf{0}$ or their weighted sum equal to zero. For a strongly exponential distribution the exponent can only equal a constant if $\boldsymbol{\theta} = \mathbf{0}$ (see (65)) and therefore the derivative can also only be 0 if $\boldsymbol{\theta} = \mathbf{0}$. From Lemma 1 we have that the components of the sufficient statistic can not be written as a function of each other. Therefore it can be seen that $\mathbf{T}'(x) \neq \mathbf{0}$ and even $T_i'(x) \neq 0$. Because, if any $T_i'(x)$ was equal to zero the corresponding θ_i could be an arbitrary number different from zero while the rest of $\boldsymbol{\theta}$ is zero and thus $\boldsymbol{\theta} \neq \mathbf{0}$ but the derivative would equal zero. which violates the statement that the distribution is strongly exponential. Thus, we may conclude that for a strongly exponential distribution $T_i'(x)$ must be different from zero.

C. LEMMA 3

Consider a strongly exponential distribution of size $k \geq 2$ with sufficient statistic $\mathbf{T}(x) = (T_1(x), \ldots, T_k(x))$. Further assume that $\mathbf{T}$ is differentiable almost everywhere. Then there exist k distinct values x_1 to x_k such that $(\mathbf{T}'(x_1), \ldots, \mathbf{T}'(x_k))$ are linearly independent in $\mathbb{R}^k$

Recall that in order for the distribution to be strongly exponential then the only choice of parameter that can lead to the exponent being constant for all x is $\boldsymbol{\theta} = \mathbf{0}$. Since both $\mathbf{T'(x)}$ and $\boldsymbol{\theta}$ is in $\mathbb{R}^k$ this necessarily means that $\mathbf{T'(x)}$ must be able to span the full $\mathbb{R}^k$. That is, there exists at least k vectors of $\mathbf{T'(x)}$ in k points $x_1, \ldots, x_k$ such that the matrix $B = [\mathbf{T}'(x_1)\ \mathbf{T}'(x_2)\ \ldots\ \mathbf{T}'(x_k)]$ has full rank:

$$\begin{aligned} Rank(B) &= Rank(\begin{bmatrix}\mathbf{T}'(x_1) & \mathbf{T}'(x_2) & \ldots & \mathbf{T}'(x_k)\end{bmatrix}) \\ &= k \end{aligned} \tag{69}$$

If $Rank(B) \neq k$ then the nullity of A will be greater than 1 and thus any vector from the orthogonal complement of the column space of B can be picked as $\boldsymbol{\theta}^*$ such that $\boldsymbol{\theta}^* \neq \mathbf{0}$ and $\langle \mathbf{T(x)'}, \boldsymbol{\theta}^* \rangle = 0$ for all x. However, if that is the case then the distribution is not strongly exponential as seen from Lemma 2 that shows that only a distribution which is *not* strongly exponential will have $\langle \mathbf{T(x)'}, \boldsymbol{\theta} \rangle = 0$ for $\boldsymbol{\theta} \neq \mathbf{0}$. Therefore, in a strongly exponential distribution there must exist k points, $x_1, \ldots, x_k$ such that the column vectors of A are linearly independent.

These three Lemmas have been used to derive the important property that in univariate exponential distributions which are minimal in k and strongly exponential there exist at least k points, $x_1, \ldots, x_k$ such that the vectors $\mathbf{T}'(x_1), \ldots, \mathbf{T}'(x_k)$ are linearly independent. We can now use this to show that the $nk \times nk$ matrix A in (70) is invertible under the assumption that the $nk \times n$ Jacobian matrix of $\mathbf{T}(\mathbf{f}^{-1}(x))$, $J_{\mathbf{T}}(\mathbf{f}^{-1}(x))$, exists and is of rank n:

$$\mathbf{T}(\mathbf{f}^{-1}(\mathbf{x})) = A\tilde{\mathbf{T}}(\tilde{\mathbf{f}}^{-1}(\mathbf{x})) + \mathbf{c} \tag{70}$$

To make the proof easier to follow we examine the form the Jacobian matrix will have. First we write the expression for the Jacobian matrix of $\mathbf{T}(\mathbf{f}^{-1}(x))$ (remembering that $\mathbf{f}^{-1}$ is a function that maps x to $\mathbb{R}^n$, such that $\mathbf{T}$ is a function of n (latent) variables, $f_1^{-1}(x), \ldots, f_n^{-1}(x)$: (71)–(73), as shown at the bottom of the next page.

We can notice that this matrix will have a particular shape since many of the entries will become zero as they are not a function of the variable with which the derivative is taken. Therefore the Jacobian matrix will have the shape:

$$J_{\mathbf{T}}(\mathbf{f}^{-1}(x)) = \begin{bmatrix} T_{1,1}'(f_1^{-1}(x)) & 0 & \ldots & 0 \\ \vdots & \vdots & & \vdots \\ T_{1,k}'(f^{-1}(x)) & 0 & \ldots & 0 \\ 0 & T_{2,1}'(f_2^{-1}(x)) & \ldots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & T_{2,k}'(f_2^{-1}(x)) & \ldots & 0 \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \ldots & T_{n,1}'(f_n^{-1}(x)) \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \ldots & T_{n,k}'(f_n^{-1}(x)) \end{bmatrix} \tag{74}$$

The expression we used to derive invertibility of A for $k = 1$ was found in (63) as seen below:

$$J_{\mathbf{T}}(x) = AJ_{\tilde{\mathbf{T}}\circ\tilde{\mathbf{f}}^{-1}}(x) \tag{75}$$

For $k = 1$ we used the fact that $J_{\mathbf{T}}(x)$ becomes an $n \times n$ matrix of rank n to prove A is invertible. The same approach is used now, but since for $k > 1$ $J_{\mathbf{T}}(x)$ is an $nk \times n$ matrix which is not square and thus not invertible we use that the above expression is true for all x. In particular, we choose k points $x_1, \ldots, x_k$ according to Lemma 3. The k points serve two purposes. First, k Jacobians are needed to have enough entries to fill an $nk \times nk$ square matrix of Jacobians and secondly by choosing the points according to Lemma 3 invertibility can be proved. By concatenating the Jacobian matrices evaluated at the k points the matrix Q can be formed:

$$Q = \begin{bmatrix} & | & & | & \\ J_{\mathbf{T}}(x_1) & | & \ldots & | & J_{\mathbf{T}}(x_k) \\ & | & & | & \end{bmatrix} \tag{76}$$

And similarly for $J_{\tilde{\mathbf{T}}\circ\tilde{\mathbf{f}}^{-1}}(x)$:

$$\tilde{Q} = \begin{bmatrix} & | & & | & \\ J_{\tilde{\mathbf{T}}\circ\tilde{\mathbf{f}}^{-1}}(x_1) & | & \ldots & | & J_{\tilde{\mathbf{T}}\circ\tilde{\mathbf{f}}^{-1}}(x_k) \\ & | & & | & \end{bmatrix} \tag{77}$$

Thus for $k \geq 1$ we may write:

$$Q = A\tilde{Q} \tag{78}$$

To verify that the concatenated system can indeed be written as in (78) we can recognize the expression as block matrix multiplication where A is a matrix with $q = 1$ row partition and $s = 1$ column partition and $\tilde{Q}$ a block matrix with $s = 1$ row partition and $r = k$ column partitions. Thus, the partitions of A and $\tilde{Q}$ are conformable and the resulting matrix, Q, will have 1 row partition and k column partitions, as in the definition above. We may also confirm that the new matrix multiplication maps every column partition i in $\tilde{Q}$ to column partition i in Q as it should be from (75). Block matrix multiplication is defined as:

$$Q_{qr} = \sum_{i=1}^{s} A_{qi}\tilde{Q}_{ir} = A_{11}\tilde{Q}_{1r} = A\tilde{Q}_{1r} = Q_{1r} \tag{79}$$

Therefore each column partition of $\tilde{Q}$ is mapped to the same column partition in Q by A. Every column partition in Q has the form of (74) and by rearranging the columns of Q such that all the nonzero elements are grouped it can be seen that Q can be written as a block diagonal matrix:

$$Q = \begin{bmatrix} B_{f_1^{-1}} & & & 0 \\ & B_{f_2^{-1}} & & \\ & & \ddots & \\ 0 & & & B_{f_n^{-1}} \end{bmatrix} \tag{80}$$

$$J_{\mathbf{T}}(\mathbf{f}^{-1}(x)) = \frac{\mathrm{d}\mathbf{T}(\mathbf{f}^{-1}(x))}{\mathrm{d}(f_1(x), f_2(x), \ldots, f_n(x))} \tag{71}$$

$$= \begin{bmatrix} \frac{\mathrm{d}\mathbf{T}(\mathbf{f}^{-1}(x))}{\mathrm{d}f_1(x)} & \frac{\mathrm{d}\mathbf{T}(\mathbf{f}^{-1}(x))}{\mathrm{d}f_2(x)} & \ldots & \frac{\mathrm{d}\mathbf{T}(\mathbf{f}^{-1}(x))}{\mathrm{d}f_n(x)} \end{bmatrix} \tag{72}$$

$$= \begin{bmatrix} \frac{\mathrm{d}T_{1,1}(f_1^{-1}(x))}{df_1^{-1}(x)} & \frac{\mathrm{d}T_{1,1}(f_1^{-1}(x))}{df_2^{-1}(x)} & \ldots & \frac{\mathrm{d}T_{1,1}(f_1^{-1}(x))}{df_n^{-1}(x)} \\ \vdots & \vdots & & \vdots \\ \frac{\mathrm{d}T_{1,k}(f_1^{-1}(x))}{df_1^{-1}(x)} & \frac{\mathrm{d}T_{1,k}(f_1^{-1}(x))}{df_2^{-1}(x)} & \ldots & \frac{\mathrm{d}T_{1,k}(f_1^{-1}(x))}{df_n^{-1}(x)} \\ \frac{\mathrm{d}T_{2,1}(f_2^{-1}(x))}{df_1^{-1}(x)} & \frac{\mathrm{d}T_{2,1}(f_2^{-1}(x))}{df_2^{-1}(x)} & \ldots & \frac{\mathrm{d}T_{2,1}(f_2^{-1}(x))}{df_n^{-1}(x)} \\ \vdots & \vdots & & \vdots \\ \frac{\mathrm{d}T_{2,k}(f_2^{-1}(x))}{df_1^{-1}(x)} & \frac{\mathrm{d}T_{2,k}(f_2^{-1}(x))}{df_2^{-1}(x)} & \ldots & \frac{\mathrm{d}T_{2,k}(f_2^{-1}(x))}{df_n^{-1}(x)} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ \frac{\mathrm{d}T_{n,1}(f_n^{-1}(x))}{df_1^{-1}(x)} & \frac{\mathrm{d}T_{n,1}(f_n^{-1}(x))}{df_2^{-1}(x)} & \ldots & \frac{\mathrm{d}T_{n,1}(f_n^{-1}(x))}{df_n^{-1}(x)} \\ \vdots & \vdots & & \vdots \\ \frac{\mathrm{d}T_{n,k}(f_n^{-1}(x))}{df_1^{-1}(x)} & \frac{\mathrm{d}T_{n,k}(f_n^{-1}(x))}{df_2^{-1}(x)} & \ldots & \frac{\mathrm{d}T_{n,k}(f_n^{-1}(x))}{df_n^{-1}(x)} \end{bmatrix}^{nk \times n} \tag{73}$$

where B is defined as in Lemma 3, $B = [\mathbf{T}'(x_1)\ \ \mathbf{T}'(x_2)\ \ \ldots\ \ \mathbf{T}'(x_k)]$, and the subscript f_i^{-1} is used to emphasize $\mathbf{T(f^{-1}(x))}$ is a function of n variables, $f_1^{-1}(x), \ldots, f_n^{-1}(x)$, such that $B_{f_i^{-1}}$ is the $k \times k$ matrix containing the nonzero derivatives with respect to f_i^{-1} as seen in (74) for all k points $x_1, \ldots, x_k$. If Q is invertible that would imply the invertibility of both A and $\tilde{Q}$ as it can be seen from (78) which is what we would like to prove. A block diagonal matrix is invertible if all the diagonal matrices are invertible. That is:

$$Q^{-1} = \begin{bmatrix} B_{f_1^{-1}} & & & 0 \\ & B_{f_2^{-1}} & & \\ & & \ddots & \\ 0 & & & B_{f_n^{-1}} \end{bmatrix}^{-1} = \begin{bmatrix} B_{f_1^{-1}}^{-1} & & & 0 \\ & B_{f_2^{-1}}^{-1} & & \\ & & \ddots & \\ 0 & & & B_{f_n^{-1}}^{-1} \end{bmatrix} \tag{81}$$

Since the points $x_1, \ldots, x_k$ are chosen as in Lemma 3 every diagonal matrix of Q is exactly identical to B in Lemma 3 (one for each of the n latent variables). Therefore every diagonal matrix of Q is invertible because B has full rank, as proven in Lemma 3, and thus Q is also invertible. Since Q is invertible we can write:

$$Q^{-1} = (A\tilde{Q})^{-1} = \tilde{Q}^{-1}A^{-1} \tag{82}$$

Which means that both A and $\tilde{Q}$ are invertible. Since A is invertible we have proven the equivalence relation for $k \geq 1$.

APPENDIX. EXPERIMENT DETAILS

In this section we will provide the experimental setup used to achieve the reported results. To monitor training, we logged metrics of interest such as loss, MCC, percentage of how many generated images were classified as real etc…every n iterations. In addition, images of the generated output were saved along with plots of the magnitude of the gradients at each hidden dimension. To visualize and draw meaningful conclusion from all the logged data we created an interactive plot that simultaneously show interactive graphs of the logged data, meaning all graphs can be zoomed, dragged, scaled etc. and individual points can be inspected at cursor hover as well as the saved images. Since all datapoints are associated with an iteration and all images are also associated with an iteration whenever the mouse hovers a datapoint the plot is updated to show the images and gradients of that particular iteration as it can be seen in FIGURE 4 and 5.

A. HYPERPARAMETER TUNING

Adversarial training is known to require extensive hyperparameter tuning since there needs to be a balance between the generator and discriminator. If either is too complex or simple the training will collapse. Our approach was to use the same decoder size as the iVAE model because it would allow direct comparison of the two models and it was clear that the iVAE decoder was sufficiently complex to represent the data faithfully. In this way the hyperparameter tuning was simplified to finding an adequate discriminator. By sweeping over different discriminator sizes a discriminator size that would not collapse training and produce faithful reconstruction of the data could be found for each number of observations per segment. The found discriminator sizes are those seen in TABLE 1. Batch size and discriminator size are also closely

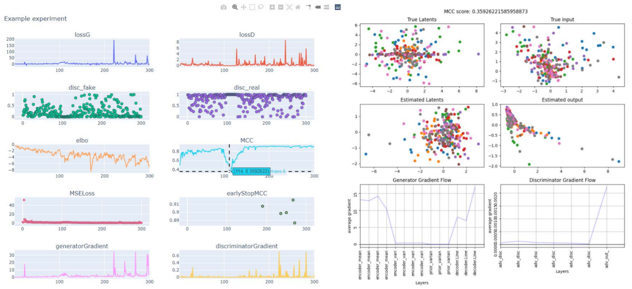

FIGURE 4. Interactive plots with updating images on hover.

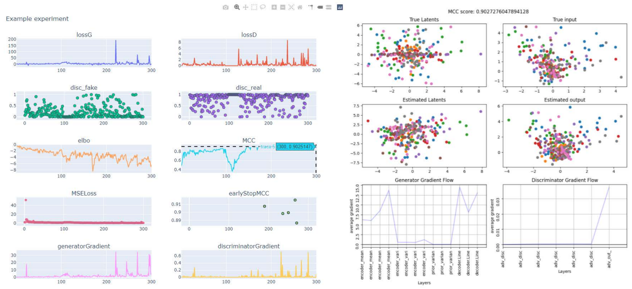

FIGURE 5. Interactive plots with updating images on hover.

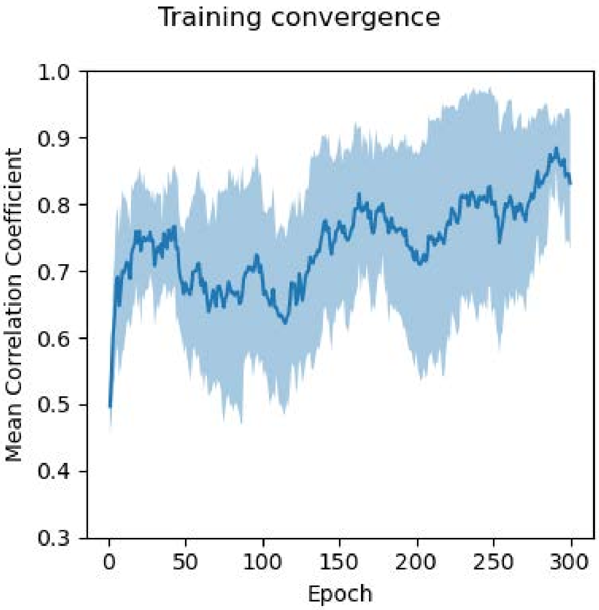

FIGURE 6. Statistics of the MCC convergence across seeds with standard deviation error bars included. (Epochs in thousands).

TABLE 2. Data generator parameters.

Data dimension	Number of segments	Number of observations per segment	Mixing layers
2	5	{100, 200, 500, 1000, 2000}	3

related as the batch size determines how many samples is presented to the discriminator and thus larger batch sizes tend to require a larger discriminator. We found no trivial tendency such that e.g. when the batch size is doubled so should the size of the discriminator be. In practice a batch size that seemed appropriate for the used device was chosen and then the above-mentioned sweep over discriminator size was performed.

B. DATA GENERATION

The used data was generated using the same data generator as in [1], [2]. The parameters of the data generator is summarized in TABLE 2:

The same parameters are used for the generated data in all three models.

REFERENCES

[1] I. Khemakhem, R. Monti, D. Kingma, and A. Hyvarinen, "ICE-BeeM: Identifiable conditional energy-based deep models based on nonlinear ICA," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 12768–12778.

[2] I. Khemakhem, D. Kingma, R. Monti, and A. Hyvarinen, "Variational autoencoders and nonlinear ICA: A unifying framework," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 2207–2217.

[3] H. Trygve, "The probability approach in econometrics," *Economet.*, vol. 12, pp. iii–115, Jan. 1944. [Online]. Available: http://www.jstororg/stable/1906935

[4] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4114–4124.

[5] A. Hyvärinen and P. Pajunen, "Nonlinear independent component analysis: Existence and uniqueness results," *Neural Netw.*, vol. 12, no. 3, pp. 429–439, Apr. 1999.

[6] P. Sorrenson, C. Rother, and U. Köthe, "Disentanglement by nonlinear ICA with general incompressible-flow networks (GIN)," 2020, *arXiv:2001.04872*.

[7] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014.

[8] A. B. L. Larsen, S. K. Sonderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proc. 33rd Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, vol. 48, M. F. Balcan and K. Q. Weinberger, Eds. New York, NY, USA, Jun. 2016, pp. 1558–1566. [Online]. Available: https://proceedings.mlr.press/v48/larsen16.html

[9] B. Sriperumbudur, K. Fukumizu, A. Gretton, A. Hyvärinen, and R. Kumar, "Density estimation in infinite dimensional exponential families," *J. Mach. Learn. Res.*, vol. 18, pp. 1830–1888, Jul. 2017.

[10] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *Stat*, vol. 1050, p. 1, May 2014.

[11] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[12] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016.

[13] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of Wasserstein GANs," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5769–5779.

[14] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 214–223.

[15] K. Xu, C. Li, J. Zhu, and B. Zhang, "Understanding and stabilizing GANs' training dynamics using control theory," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 10566–10575.

[16] S. Jenni and P. Favaro, "On stabilizing generative adversarial training with noise," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12137–12145.

BJØRN UTTRUP DIDERIKSEN received the B.Sc. degree in electronics and IT and the M.Sc. degree in signal processing and computing from Aalborg University, Denmark, in 2019 and 2021, respectively. His research interests include machine learning, identifiability, and disentanglement.

KRISTOFFER DEROSCHE received the B.Sc. degree in electronics and IT and the M.Sc. degree in signal processing and computing from Aalborg University, Denmark, in 2019 and 2021, respectively. His research interests include machine learning, disentanglement, scientific computing, and optimization.

ZHENG-HUA TAN (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from Hunan University, Changsha, China, in 1990 and 1996, respectively, and the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University (SJTU), Shanghai, China, in 1999. He was a Visiting Scientist at the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, USA, an Associate Professor at the Department of Electronic Engineering, SJTU, and a Postdoctoral Fellow at the AI Laboratory, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea. He is a Professor with the Department of Electronic Systems and the Co-Head of the Centre for Acoustic Signal Processing Research, Aalborg University, Aalborg, Denmark. He is also the Co-Lead of the Pioneer Centre for AI, Denmark. He has coauthored over 200 refereed publications. His research interests include machine learning, deep learning, pattern recognition, speech and speaker recognition, noise-robust speech processing, multimodal signal processing, and social robotics. He was the General Chair of the IEEE MLSP 2018 and a TPC Co-Chair of the IEEE SLT 2016. He is the Chair of the IEEE Signal Processing Society Machine Learning for Signal Processing Technical Committee (MLSP TC). He is an Associate Editor of the IEEE/ACM Transactions on Audio, Speech and Language Processing. He has served as an Associate/Guest Editor for several other journals.