Discovery of Single Independent Latent Variable

Anonymous Author(s)
Affiliation
Address
email

Abstract

Latent variable discovery is a central problem in data analysis with a broad range of applications in applied science. In this work, we consider data given as an invertible mixture of two statistically independent components, and assume that one of the components is observed while the other is hidden. Our goal is to recover the hidden component. For this purpose, we propose an autoencoder equipped with a discriminator. Unlike the standard nonlinear ICA problem, which was shown to be non-identifiable, in the special case of ICA we consider here, we show that our approach can recover the component of interest up to entropy-preserving transformation. We demonstrate the performance of the proposed approach on several datasets, including image synthesis, voice cloning, and fetal ECG extraction.

1 Introduction

The recovery of hidden components in data is a long-standing problem in applied science. This problem dates back to the classical PCA [46, 22], yet it has numerous modern manifestations and extensions, e.g., kernel-based methods [49], source separation [13, 4], manifold learning [56, 48, 3, 11], and latent Dirichlet allocation [6], to name but a few. Perhaps the most relevant line of work in the context of this paper is independent component analysis (ICA) [25], which attempts to decompose an observed mixture into statistically independent components.

Here, we consider the following ICA-related recovery problem. Assume that the data is generated as an invertible mixture of two (not necessarily one dimensional) independent components, and that one of the components is observed while the other is hidden. In this setting, our goal is to recover the latent component. At first glance, this problem setting may seem specific and perhaps artificial. However, we posit that it is in fact broad and applies to many real-world problems.

For example, consider thorax and abdominal electrocardiogram (ECG) signals measured during labor for the purpose of determining the fetal heart activity. In analogy to our problem setting, the abdominal signal can be viewed as a mixture of the maternal and fetal heart activities, the maternal signal can be viewed as an accurate proxy of the maternal heart activity alone, and the fetal heart activity is the hidden component of interest we wish to recover. In another example from a different domain, consider a speech signal as a mixture of two independent components: the spoken text and the speaker identity. Arguably, the speaker identity is associated with the pitch and timbre, which is independent from information about the textual content, rhythm and volume. Consequently, recovering a speaker-independent representation of the spoken text facilitates speech synthesis and voice conversion.

In this paper, we present an autoencoder-based approach, augmented with a discriminator, for this recovery problem. First, we theoretically show that this architecture of solution facilitates the recovery of the latent component up to an entropy-preserving transformation. Second, in addition to the recovery of the latent component (the so-called analysis task), it enables us to generate new

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

mixtures corresponding to new instances of the observed independent component (the so-called synthesis task). Experimentally, we show both analysis and synthesis results on several datasets, consisting of simulated and real-world data. In particular, we demonstrate the proposed approach on ECG analysis, image synthesis, and voice cloning tasks.

Our contributions are as follows. (i) We propose an easy-to-train mechanism for extraction of a single latent independent component. (ii) We present a simple proof for the ability of the proposed approach to recover the latent component. (iii) We experimentally demonstrate the applicability of the proposed approach in the contexts of both analysis and synthesis tasks. Specifically, we show applications to real-world data from different domains.

2 Related Work

The problem we consider in this work could be viewed as a simplified case of the classical formulation of nonlinear ICA. Several algorithms have been proposed for recovery of the independent components, assuming that (i) the mixing of the components is linear, and (ii) the components (with a possible exception of one) are non-Gaussian; this case was proven to be identifiable [12, 15]. The nonlinear case, however, i.e., when the mixing of the independent components is an arbitrary invertible function, was proven to be non-identifiable in the general case [26].

Identifiable nonlinear ICA. Hyvarinen et al. [27] have recently described a general framework for identifiable nonlinear ICA, generalizing several earlier identifiability results for time series, e.g., [23, 24, 53], in which in addition to observing the data x, the framework requires an auxiliary observed variable u, so that conditioned on u, the latent factors are independent (i.e., in contrast to being marginally independent as in a standard ICA setting). The approach we propose in this work falls into this general setting, as we assume that the auxiliary variable u is in fact one of the latent factors, which immediately satisfies the conditional independence requirement. For this special case we provide a simple and intuitive recovery guarantee.

The following works have recently extended the framework of Hyvarinen et al. [27] to generative models [31], unknown intrinsic problem dimension [52], and multiview setting [19].

With respect to iVAE [31], we consider a special case where the auxiliary u is part of the mix and equals one of the latent components. However, we allow for the recovery of high-dimensional components, whereas in iVAE, only one-dimensional components are recovered. In addition, our work presents several important differences: (i) we formulate our recovery guarantee result in terms of entropy-preserving map rather than statistical identifiability. (ii) It allows for a compact proof of the recovery guarantee. (iii) We present experiments on real-world data and comparisons to leading methods per application domain, whereas in [31], the iVAE approach was mostly demonstrated on simulated data.

Disentangled representation learning methods. While the main interest in ICA has originally been for purposes of analysis (i.e., recovery of the independent sources from the observed data), the highly impressive achievements in deep generative modeling over recent years have drawn much interest in the other direction, i.e., synthesis of data (e.g., images) from independent factors. In the research community, this direction is often termed learning of disentangled representations, i.e., representations in which modification of a single latent coordinate in the representation affects the synthesized data by manipulating a single perceptual factor in the observed data, leaving other factors unchanged. In a similar fashion to the ICA case, the task of learning disentangled representations in the general case was proved to be non-identifiable [41].

Several methods for learning disentangled representations have been recently proposed. However, they lack identifiability guarantees, as most of which are based on a variational autoencoder (VAE, Kingma and Welling [34]) formulation, and decompositions of the VAE objective, for example [21, 32, 10, 9, 35, 8, 16]. GAN-based approaches for disentanglement have been proposed as well [10, 7].

Domain confusion. Our proposed approach is based on the ability to learn an encoding in which different conditions (i.e., states of the observed factor) are indistinguishable. Such principle has been in wide use in the machine learning community for domain adaptation tasks [17]. A popular means to achieve this approximate invariance is via discriminator networks, for example, in [37, 44, 50, 45], where the high-level mechanism is similar to the one proposed in this work, although the specifics are different.

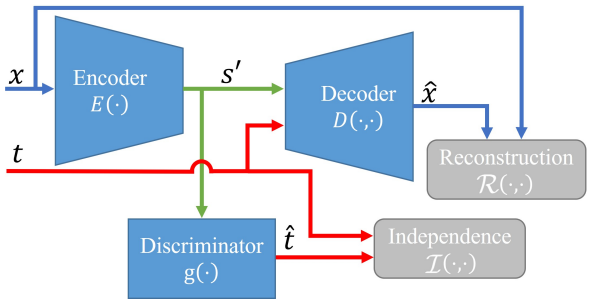

Figure 1: A diagram of the proposed approach. Learned functions are colored in blue, while the objective functions are colored in gray.

We remark that while the algorithm we propose here has been presented before, e.g., in the context of singing voice conversion [45], to the best of our knowledge, it was not discussed in the context of latent independent component recovery, and no identifiability results were shown.

3 Problem Formulation

Let $X \in \mathbb{R}^d$ be a random variable. We assume that X is generated via $X = f(S, T)$, where f is an unknown invertible function of two arguments, and $S \in \mathcal{S}$ and $T \in \mathcal{T}$ are two random variables in arbitrary domains, satisfying $S \perp\!\!\!\perp T$. We refer to S as the *unobserved source* that we wish to recover, to T as the *observed condition*, and to X as the *observed input*.

Let $\{(s_i, t_i)\}_{i=1}^n$ be n realizations of $S \times T$, and for all $i = 1, \ldots, n$, let $x_i = f(s_i, t_i)$. Given only input-condition pairs, i.e., $\{(x_i, t_i)\}_{i=1}^n$, we state two goals. First, from analysis standpoint, we aim to recover the realizations $s_1, \ldots s_n$ of the unobserved source S. Second, from synthesis standpoint, for any (new) realization t of T, we aim to to generate an instance $x = f(s_i, t)$ for any $i = 1, \ldots, n$.

4 Autoencoder Model with Discriminator

To achieve the above goals, we propose an autoencoder (AE) with a discriminator. The AE model is denoted by (E, D), where the encoder E maps inputs x to codes s' (analysis), and the decoder D maps code-condition tuples (s', t) to input reconstructions $\hat{x}$ (synthesis), i.e., $x \overset{E}{\longmapsto} s'$ and $(s', t) \overset{D}{\longmapsto} \hat{x}$. The discriminator, denoted by $g(\cdot)$, maps codes s' to predicted conditions $\hat{t}$[1].

We use two objective terms in a GAN-like minmax game:

$$\min_{E,D} \max_{g} \left[\mathcal{R}\left(x, D(E(x), t)\right) - \lambda \mathcal{I}\left(t, g(E(x))\right)\right]. \tag{1}$$

The first objective, denoted by $\mathcal{R}$, measures the discrepancy between an input x and its reconstruction $\hat{x} = D(E(x), t)$. The second objective, denoted by $\mathcal{I}$, quantifies the independence between the condition t and code s' via a prediction $\hat{t} = g(s')$. In order to maximize $-\mathcal{I}\left(t, g(E(x))\right)$, the discriminator aims at leveraging any information in the code $s' = E(x)$ on the condition t (via a learned function g) to obtain an accurate prediction $\hat{t} = g(s')$. In order to minimize both $\mathcal{R}\left(x, D(E(x), t)\right)$ and $-\mathcal{I}\left(t, g(E(x))\right)$, the autoencoder aims at reconstructing the input from the code s' and the condition t, while failing the discriminator. We will show formally and empirically, that this results in an equilibrium in which s' does not contain any information on t and contains all the remaining information in x. Our approach is illustrated in Figure 1.

4.1 Recovery of the Latent Component

Let $S' = E(X)$ be a random variable representing the encoder output, and let $\hat{X} = D(S', T)$ be a random variable representing the decoder output.

[1] This formulation of the discriminator does not capture all scenarios, however we use it here for simplicity. In Section 4.2, we describe additional implementations of the discriminator.

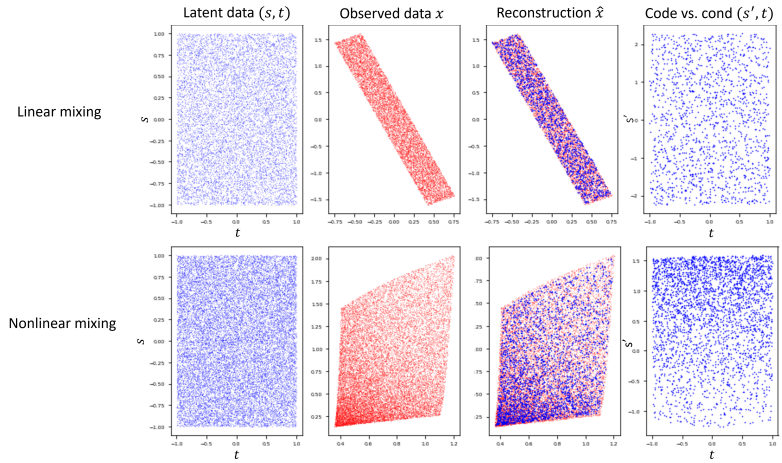

Figure 2: Analysis demonstration. The reconstruction plots show train data in red and reconstructed test data in blue. The learned code s' is a recovery of the latent factor s, up to entropy-preserving transformation (e.g., an arbitrary monotonic transformation). The approximate independence of the S' and the condition T can be recognized by noticing that the joint density of the code and the condition is an outer product of the marginal distributions. The p-values of χ^2 independence test for the shown results are 0.75 (linear mixing) and 0.83 (nonlinear mixing)

Lemma 4.1 establishes that when the autoencoder is trained to perfect reconstruction and the learned code is independent of the condition, the learned code contains the same information as S, thus proving that the latent component of interest can be recovered, up to an entropy-preserving transformation. The lemma is stated in terms of mutual information $I(\cdot;\cdot)$ and differential entropy $h(Y) := -\int_{y \in \mathcal{Y}} p(y) \log p(y) dy$, where Y is a random variable with density p, taking values in $\mathcal{Y}$.

Lemma 4.1. *Assume S is continuous. Suppose we train the autoencoder model to zero generalization loss, i.e., $\hat{X} = X$, and impose on the code that $S' \perp\!\!\!\perp T$. Then $I(S; S') = h(S) = h(S')$.*

Proof. Since $S \perp\!\!\!\perp T$, $h(S|S') = h(S, T|S', T)$. Since $X = f(S, T)$ and f is invertible, $\Pr(X = x, S = s, T = t) = \Pr(S = s, T = t) = \Pr(X = x)$ whenever $x = f(s, t)$ and 0 otherwise, where for brevity, with a slight abuse of notation we have used $\Pr$ to denote densities. Then $h(S, T|S', T) = h(X|S', T)$. Since $\hat{X} = X$ we have

$$h(X|S', T) = h(\hat{X}|S', T) = h(\hat{X}, S', T) - h(S', T) = 0.$$

Therefore $h(S|S') = 0$, and $I(S; S') = h(S) - h(S|S') = h(S)$. Similarly, $I(S'; S) = h(S') - h(S'|S) = h(S')$. □

Lemma 4.1 has two important consequences. First, it shows that unlike the standard nonlinear ICA problem, the problem we consider here allows for recovery of the latent independent component of interest. More specifically, it proves that when the autoencoder yields perfect reconstruction and condition-independent code, the learned code is a recovery of the random variable S, up to entropy-preserving transformation. Second, the lemma prescribes a recipe for the practical solution we present here. Requiring the autoencoder to generate accurate reconstruction $\hat{X}$ of X ensures that no information on S is lost in the encoding process. Independence of S' and T is achieved implicitly; it results from the equilibrium of the GAN-like minmax game (1), as the discriminator can benefit from any mutual information between S' and T.

4.2 Training Objectives

As described above, to obtain a code S', which is independent of the condition T, we utilize a discriminator network, aiming to leverage information on T in S' for prediction, and train the encoder E to fail the discriminator, in a standard adversarial training fashion. Doing so pushes the learned codes S' towards being a condition-free encoding of X.

We propose to optimize the following objectives for the discriminator and autoencoder:

$$\mathcal{L}_{\text{disc}} = \min_{g} \mathcal{I}\left(t, g(E(x))\right) \tag{2}$$

$$\mathcal{L}_{\text{AE}} = \min_{E,D} \left[\mathcal{R}\left(x, D(E(x), t)\right) - \lambda \mathcal{I}\left(t, g(E(x))\right)\right], \tag{3}$$

The specific reconstruction and independence objective terms are application-dependent. In our experiments, we make use of the following.

Reconstruction. We use standard reconstruction loss functions. In the experiments with images, ℓ_1 and SSIM loss [58] (and combinations of these) are used. ℓ_1 loss is also used in the audio experiments, and MSE loss in the experiments with ECG signals.

Independence. The discriminator computes a map $\hat{t} = g(s')$, where $s' = E(x)$ is the code obtained from the encoder and $\hat{t} = g(s')$ is the condition predicted by the discriminator. The discriminator is trained to minimize the independence term $\mathcal{I}(\hat{t}, t)$ (thus to leverage mutual information in S' and T).

When the condition takes values from a finite symbolic set, we train the discriminator as a classifier that predicts the condition class from the code, and we set the independence term to $\mathcal{I}(\hat{t}, t) =$ Cross Entropy$(\hat{t}, t)$. This is also known as a Domain Confusion term. Using this term, the autoencoder is trained to produce codes that maximize the cross entropy with respect to the true condition, and the equilibrium of the game is when this term equals the cross entropy of a random guess.

When the condition and its prediction take numerical values, i.e., $t, \hat{t} \in \mathbb{R}$, we train the discriminator as a regression model and set the independence term to: $\mathcal{I}(\hat{t}, t) = -\text{Correl}^2(\hat{t}, t)$. We remark that this term also equals the R^2 term of a simple regression model, regressing t on $\hat{t}$. Using this term, the autoencoder is trained to produce codes for which the squared correlation with the condition is minimized. As $\hat{t}$ is a nonlinear function of s', computed via a flexible model such as a neural net, $\left(\text{Correl}(\hat{t}, t)\right)^2 = 0$ implies that S' and T are approximately statistically independent.

In addition, we also successfully train the discriminator in a contrastive fashion, i.e., to distinguish between "true tuples" (s', t) that correspond to samples (s, t) from the joint distribution of S and T satisfying $s' = E(x)$ and $x = f(s, t)$, and "fake tuples" (s', t), where $s' = E(x)$ but $x = (s, \tilde{t})$ with $t \neq \tilde{t}$. The contrastive objective is then $\mathcal{I}(s', t) =$ Cross Entropy$(\hat{l}, l)$, where l is the ground truth true/fake label and $\hat{l} = g(s', t)$ is the predicted true/fake label made by the discriminator. Note that this implementation deviates from the discriminator formulation we used thus far. Here, the discriminator $g(\cdot, \cdot)$ maps tuples of codes and conditions (s', t) to true/fake labels.

We remark that other possible implementations of the independence criterion can be utilized as well, e.g., nonlinear CCA [2, 43] and the Hilbert-Schmidt Independence Criterion (HSIC) [20].

Optimizers. Our proposed approach utilizes two optimizers, one for the autoencoder and one for the discriminator. The AE optimizer optimizes $\mathcal{L}_{\text{AE}}$ by tuning the encoder and decoder weights. The discriminator optimizer optimizes $\mathcal{L}_{\text{disc}}$ by tuning the discriminator weights (which determine the function g). Common practice in training GANs is to call the two optimizers with different frequencies. We specify the specific choices used in our experiments in Appendix C.

GAN real/ fake discriminator. Optionally, a GAN-like real / fake discriminator can be added as an additional discriminator in order to encourage generating more realistic inputs. While we have a successful empirical experience with such GAN discriminators (e.g., see Appendix B), this is not a core requirement of our proposed approach.

5 Experimental Results

In this section, we demonstrate the efficacy of the proposed approach in various settings, by reporting experimental results obtained on different data modalities and condition types, in both analysis and synthesis tasks. We present four applications here and additional two in the appendix.

We begin with a two dimensional analysis demonstration, in which the condition is real-valued. Second, we demonstrate the usage of our approach for image manipulation, where the condition is given as an image. Third, the proposed approach is used for voice cloning, which is primarily a synthesis task with symbolic condition. Fourth, we apply our approach to an ECG analysis task, using

Figure 3: Experiment setup of the rotating figures. Bulldog and Bunny rotate in different speeds. The right camera captures both Bulldog and Bunny, while the left camera captures only Bulldog. Images from the right camera are considered as the input x, which is generated from two independent factors – the rotation angles of the figures. Bulldog is considered as the condition t. The goal is to recover the rotation angle of Bunny, and to manipulate a given input image x by plugging in a different condition than the one present in the image. The fact that t and x are captured from two different viewpoints prevents modification of the image simply by pasting Bulldog into x.

Figure 4: The rotating figures experiment. From left to right: (i) the input image to the encoder x, (ii) the condition t corresponding to Bulldog in x as captured from the viewpoint of the left camera, (iii) the reconstruction $\hat{x}$ of x, (iv) a new condition plugged into the decoder, and (v) the resulting manipulated image.

a real-valued heartbeat signal as the condition. Additional experimental results in image synthesis are described Appendix A and B.

The network architectures and training hyperparameters used in each of the experiments are described Appendix C. In addition, codes reproducing some of the results in this manuscript are available at `https://github.com/DiSILV`.

5.1 2D Analysis Demonstration

In this example we first generate the latent representation of the data by sampling from two independent uniform random variables. We then generate the observed data via linear mixing. We consider one of the latent components as the condition and train the autoencoder to reconstruct the observed data, while obtaining code which is independent of the condition using the regression objective.We use ℓ_1 as a reconstruction term. The top row in Figure 2 shows the latent, observed and reconstructed data, as well as the distribution of the condition and the learned code. The bottom row in Figure 2 shows the results of a similar setup, except for the mixing which is now nonlinear. As can be seen, the joint distribution of the learned code and the condition is approximately a tensor product of the marginal distributions, which implies that the latent component is indeed recovered.

5.2 Rotating Figures

In this experiment we use the setup shown in Figure 3, in which two figures, Bulldog and Bunny, rotate on discs. The rotation speeds are different and are not an integer multiple one of the other. The figures are recorded by two static cameras, where the right camera captures both Bunny and Bulldog, while the left camera captures only Bulldog. The cameras operate simultaneously, so that in each pair of images Bulldog's position with respect to the table is the same. This dataset was curated in [38].

We consider images from the right camera (which contain both figures) as the observed input x, and the images from the left camera (which only show Bulldog) as the condition t. Note that the input

Table 1: Voice cloning results: Mel Cepstral Distortion (MCD) in terms of mean (std). PPG, PPG2 results are taken from [47], VQ AVE and PPG GMM results are taken from [14].

METHOD	TTS SKINS [47]	GLE [14]	VQ VAE	PPG GMM	PPG	PPG2	OURS
MCD	8.76 (1.72)	7.56	8.43	8.57	9.19 (1.50)	9.18 (1.52)	**6.27** (1.44)

can be considered as generated from two independent sources, namely the rotation angles of Bulldog and Bunny. The goal is to use x and t to recover the rotation angle s of Bunny[2].

Once training is done, we use the autoencoder to generate new images by manipulating Bulldog's rotation angle while preserving Bunny's. This is done by feeding a (synchronous) (x, t) tuple to the encoder, obtaining an encoding s', sampling an arbitrary condition $\tilde{t}$ and feeding $(s', \tilde{t})$ through the decoder. We use ℓ_1 loss for reconstruction, and contrastive lossto train the discriminator. Namely, we train the discriminator to distinguish between (image, condition) tuples, which were shot at the same time, and tuples which were not. Figure 4 shows an exemplifying result. As can be seen, the learned model disentangles the rotation angles of Bunny and Bulldog and generates images in which Bunny's rotation angle is preserved while Bulldog's is manipulated.

5.3 Voice Cloning

To demonstrate the application of the proposed method to voice conversion, we run experiments on a non-parallel corpus, CSTR VCTK Corpus [57], which consists of 109 English speakers with several accents (e.g., English, American, Scottish, Irish, Indian, etc.). In our experiments, we use a subset of the corpus containing all the utterances for the first 30 speakers (p225- p256, without p235 and p242).

We construct the autoencoder to operate on mel spectrograms, using the speaker id as the condition. The AE architecture was based on Jasper [40] blocks (specific details can be found in Appendix C). The decoder use a learnable lookup table with 64-dimensional embedding for each speaker. For the discriminator, we use the same architecture as in [44]. We use ℓ_1 loss for reconstruction, and the discriminator is trained using domain confusion loss.Along with the reconstruction loss, in this experiment we also train a real/fake discriminator, applied to the output of the decoder. To convert the decoder output to waveform, we use a pre-trained melgan [36] vocoder.

Once the autoencoder is trained, we apply it to convert speech from any speaker to any other speaker. Some samples of the converted speech are given at `https://disilv.github.io/`. To evaluate the similarity of the converted voice and the target voice, we use MCD (Mel Cepstral Distortion) on a subset of the data containing parallel sentences of multiple speakers. Specifically, MCD computes the ℓ_1 difference between dynamically time warped instance of the converted source voice and a parallel instance of the target voice, which is a common evaluation metric for voice cloning. We remark that the parallel data are used only for evaluation and not for training the model. We use the script provided at [39] to compute the MCD and compare our proposed approach to [14, 47] and references therein, which are all considered to be strong baselines, trained on the VCTK dataset as well. The results, shown in Table 1, demonstrate that our approach outperforms these strong baselines.

5.4 Fetal ECG extraction

In this experiment we demonstrate the applicability of the proposed approach to non-invasive fetal electrocardiogram (fECG) extraction, which facilitates the important task of monitoring the fetal cardiac activity during pregnancy and labor. Following commonly-used non-invasive methods, we consider extraction of the fECG based on two signals: (i) multi-channel abdominal ECG recordings, which consist of a mixture of the desired fECG and the masking maternal electrocardiogram (mECG), and (ii) thorax ECG recordings, which are assumed to contain only the mECG. In analogy to our problem formulation (see Section 3), the desired unobserved source s denotes the fECG, the observed condition t denotes the (thorax) mECG, and the input x denotes the abdominal ECG.

[2] A related work on this dataset was done in [51], although there the goal was the opposite one, i.e., to recover the common information of the two views, which is the rotation angle of Bulldog.

Table 2: fECG extraction results. In the leftmost column, we present R_x, and in the other columns we present $R_{s'}$ achieved by the different methods.

# OF SUBJECT	INPUT	OURS	ADALINE	ESN	LMS	RLS
TOP 5	2.23 (3.23)	**6.86** (1.98)	6.46 (2.54)	1.99 (1.08)	2.60 (1.60)	1.03 (0.70)
TOP 10	1.20 (2.41)	**5.43** (2.02)	4.22 (2.94)	1.19 (1.10)	1.56 (1.53)	0.75 (0.56)
TOP 20	0.66 (1.75)	**3.53** (2.44)	2.59 (2.63)	0.71 (0.91)	0.89 (1.26)	0.51 (0.46)
ALL	0.30 (1.17)	**1.84** (2.16)	1.32 (2.08)	0.36 (0.68)	0.40 (0.94)	0.26 (0.38)

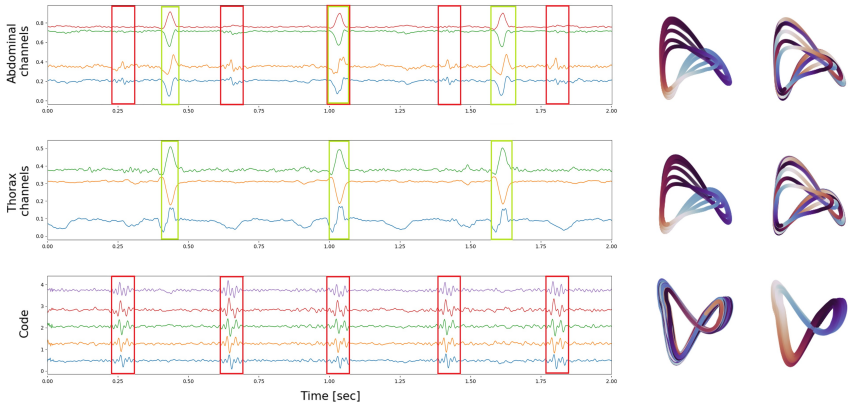

Figure 5: Example of an input-condition pair (x_i, t_i) and the obtained code s'_i. The duration of the presented time segment is 2 sec. Leftmost column: the raw channels. Middle column: the PCA embedding of the samples colored by the mECG. Rightmost column: the PCA embedding of the samples colored by the fECG. Top row: the abdominal channels x_i (for brevity only 5 channels are presented). Middle row: the thorax channels t_i. Bottom row: the obtained code s'_i.

Dataset. We consider the dataset from [54], which is publicly available[3] on PhysioNet [18]. This dataset was recently published and is part of an ongoing effort to establish a benchmark for non-invasive fECG extraction methods. The dataset consists of ECG recordings from 60 subjects. Each recording consists of $n_a = 24$ abdominal ECG channels and $n_t = 3$ thorax ECG channels. In addition, it contains a pulse-wave doppler recording of the fetal heart that serves as a ground-truth. See Appendix C.6 for more details.

Model training. The input-condition pairs (x_i, t_i) are time-segments of the abdominal ECG recordings ($x_i \in \mathbb{R}^{n_a \times n_T}$) and the thorax ECG recordings ($t_i \in \mathbb{R}^{n_t \times n_T}$), where the length of the time-segments is set to $n_T = 2,000$ (4 seconds). We train a separate model for each subject based on a collection of n input-condition pairs $\{(x_i, t_i)\}_{i=1}^n$ of time-segments.

The encoder is based on a convolutional neural network (CNN), so that the obtained codes $s'_i = E(x_i) \in \mathbb{R}^{n_d \times n_T}$ are time-segments, where the dimension of the code is set to $n_d = 5$. For more details on the architecture, model training, and hyperparameters selection, see Appendix C.6.

We note that the training is performed in an unsupervised manner, i.e., we use the ground-truth doppler signal only for evaluation and not during training.

Qualitative evaluation. In Figure 5 we present an example of an input-condition pair (x_i, t_i) and the obtained code $s'_i = E(x_i, t_i)$. We see that the abdominal channels consist of a mixture of the fECG and the mECG, where the fECG is significantly less dominant than the mECG and might even be completely absent from some of the channels. In addition, we see that the thorax channels are affected by the mECG only. Lastly, we see that the obtained code captures the fECG without any noticeable trace of the mECG. In addition, we present the projections of $1,000$ sequentially-sampled inputs x_i (abdominal channels), conditions t_i (thorax channels), and their codes $s'_i = E(x_i)$ on their respective 3 principal components. We color the projected points by the periodicity of the mECG (middle column), computed from the thorax channels, and by the periodicity of the fECG (rightmost column), computed from the ground-truth doppler signals. We see that the PCA of the abdominal and

[3] `https://physionet.org/content/ninfea/1.0.0/`

thorax channels are similar, implying that the mECG dominates the mixture. In addition, we see that the color of the PCA of the abdominal and thorax channels according to the mECG (middle column) is similar and smooth, unlike the color by the fECG (rightmost column). In contrast, the PCA of the code is different (bottom row) and only the color by the fECG is smooth (rightmost column), indicating that the code captures the fECG without a significant trace of the mECG, as desired.

Baselines. We consider four baselines taken from a recent review [29]. Specifically, we focus on methods that utilize reference thorax channels. The first two baselines are based on adaptive filtering, which is considered to be the traditional approach for fECG extraction: least mean squares (LMS) and recursive least squares (RLS). This approach was first introduced by Widrow et al. [59], and it is still considered to be relevant in recent studies [42, 60, 55]. The third baseline is ADALINE [30] which utilizes neural networks adaptable to the nonlinear time-varying properties of the ECG signal . The fourth baseline is based on echo state network (ESN) [28].

Quantitative evaluation. To the best of our knowledge, there is no gold-standard nor definitive evaluation metrics for fECG extraction. Here, based on the ground-truth doppler signal, we quantify the enhancement of the fECG and the suppression of the mECG as follows.

First, we compute the principal component of the input x_i. Second, we compute the one-sided auto-correlation of the principal component, and denote it by A_{x_i}. Then, we quantify the average presence of the fECG in the inputs x_i by computing: $\bar{A}_x^{(f)} = \frac{1}{n_s}\sum_{i=1}^{n_s} A_{x_i}(\tau_i^{(f)})$, where $\tau_i^{(f)}$ denotes the periods of the fECG obtained from the doppler signals, and n_s denotes the number of time segments in the evaluated recording. Similarly, we compute $\bar{A}_x^{(m)} = \frac{1}{n_s}\sum_{i=1}^{n_s} A_{x_i}(\tau_i^{(m)})$, where $\tau_i^{(m)}$ denotes the periods of the mECG obtained from the thorax signals. Finally, to quantify the relative presences of the signals, we compute the ratio $R_x = \frac{\bar{A}_x^{(f)}}{\bar{A}_x^{(m)}}$. We apply the same procedure to the codes s_i', resulting in $R_{s'}$. When evaluating the baselines, we consider the signals obtained after the mECG cancellation as the counterparts of our code signals.

In Table 2, we present the average ratios in the input, code, and baselines over all the subjects (see Appendix C.6 for results per subject). We note that not all the subjects in the dataset include a noticeable fECG in the abdominal recordings. Therefore, we present results over subsets of top k subjects showing highest average ratios R_x. We see that our method significantly enhances the fECG with respect to the mixture, and it outperforms the tested baselines.

6 Conclusion

In this paper, we present an autoencoder-based approach for single independent component recovery. The considered problem consists of observed data (mixture) generated from two independent components: one observed and the other hidden that needs to be recovered. We theoretically show that this ICA-related recovery problem can be accurately solved, in the sense that the hidden component is recovered up to an entropy-preserving function, by an autoencoder equipped with a discriminator. In addition, we demonstrate the relevance of the problem and the performance of the proposed solution on several tasks, involving image manipulation, voice cloning, and fetal ECG extraction.

Future research will address the limitations of this work. Lemma 4.1 assumes zero generalization loss, i.e., convergence to a global minimum, which is often not achieved in practice. In future work we plan to generalize this statement and assume bounded generalization loss. Another future direction will address noise robustness. The current setting does not consider any noise, either in t or in x, or even in f. For example, in the presence of noise, perfect reconstruction is undesired, and other losses need to be developed and used.

References

[1] AI, H. (2019). High-level expert group on artificial intelligence.

[2] Andrew, G., Arora, R., Bilmes, J., and Livescu, K. (2013). Deep canonical correlation analysis. In *International conference on machine learning*, pages 1247–1255. PMLR.

[3] Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396.

[4] Belouchrani, A., Abed-Meraim, K., Cardoso, J.-F., and Moulines, E. (1997). A blind source separation technique using second-order statistics. *IEEE Transactions on signal processing*, 45(2):434–444.

[5] Blau, Y. and Michaeli, T. (2018). The perception-distortion tradeoff. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6228–6237.

[6] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

[7] Brakel, P. and Bengio, Y. (2017). Learning independent features with adversarial nets for non-linear ica. *arXiv preprint arXiv:1710.05050*.

[8] Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. (2018). Understanding disentangling in beta-vae. *arXiv preprint arXiv:1804.03599*.

[9] Chen, R. T., Li, X., Grosse, R., and Duvenaud, D. (2018). Isolating sources of disentanglement in variational autoencoders. *arXiv preprint arXiv:1802.04942*.

[10] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *arXiv preprint arXiv:1606.03657*.

[11] Coifman, R. R. and Lafon, S. (2006). Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30.

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

[13] Comon, P. and Jutten, C. (2010). *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic press.

[14] Ding, S. and Gutierrez-Osuna, R. (2019). Group latent embedding for vector quantized variational autoencoder in non-parallel voice conversion. In *INTERSPEECH*, pages 724–728.

[15] Eriksson, J. and Koivunen, V. (2004). Identifiability, separability, and uniqueness of linear ica models. *IEEE signal processing letters*, 11(7):601–604.

[16] Esmaeili, B., Wu, H., Jain, S., Bozkurt, A., Siddharth, N., Paige, B., Brooks, D. H., Dy, J., and Meent, J.-W. (2019). Structured disentangled representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2525–2534. PMLR.

[17] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030.

[18] Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. (2000). Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220.

[19] Gresele, L., Rubenstein, P. K., Mehrjou, A., Locatello, F., and Schölkopf, B. (2020). The incomplete rosetta stone problem: Identifiability results for multi-view nonlinear ica. In *Uncertainty in Artificial Intelligence*, pages 217–227. PMLR.

[20] Gretton, A., Bousquet, O., Smola, A., and Schölkopf, B. (2005). Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77. Springer.

[21] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2016). beta-vae: Learning basic visual concepts with a constrained variational framework.

[22] Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417.

[23] Hyvarinen, A. and Morioka, H. (2016). Unsupervised feature extraction by time-contrastive learning and nonlinear ica. *arXiv preprint arXiv:1605.06336*.

[24] Hyvarinen, A. and Morioka, H. (2017). Nonlinear ica of temporally dependent stationary sources. In *Artificial Intelligence and Statistics*, pages 460–469. PMLR.

[25] Hyvärinen, A. and Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430.

[26] Hyvärinen, A. and Pajunen, P. (1999). Nonlinear independent component analysis: Existence and uniqueness results. *Neural networks*, 12(3):429–439.

[27] Hyvarinen, A., Sasaki, H., and Turner, R. (2019). Nonlinear ica using auxiliary variables and generalized contrastive learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 859–868. PMLR.

[28] Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13.

[29] Kahankova, R., Martinek, R., Jaros, R., Behbehani, K., Matonia, A., Jezewski, M., and Behar, J. A. (2019). A review of signal processing techniques for non-invasive fetal electrocardiography. *IEEE reviews in biomedical engineering*, 13:51–73.

[30] Kahankova, R., Martinek, R., Mikolášová, M., and Jaroš, R. (2018). Adaptive linear neuron for fetal electrocardiogram extraction. In *2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pages 1–5. IEEE.

[31] Khemakhem, I., Kingma, D., Monti, R., and Hyvarinen, A. (2020). Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR.

[32] Kim, H. and Mnih, A. (2018). Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR.

[33] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[34] Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[35] Kumar, A., Sattigeri, P., and Balakrishnan, A. (2017). Variational inference of disentangled latent concepts from unlabeled observations. *arXiv preprint arXiv:1711.00848*.

[36] Kumar, K., Kumar, R., de Boissiere, T., Gestin, L., Teoh, W. Z., Sotelo, J., de Brébisson, A., Bengio, Y., and Courville, A. (2019). Melgan: Generative adversarial networks for conditional waveform synthesis. *arXiv preprint arXiv:1910.06711*.

[37] Lample, G., Zeghidour, N., Usunier, N., Bordes, A., Denoyer, L., and Ranzato, M. (2017). Fader networks: Manipulating images by sliding attributes. *arXiv preprint arXiv:1706.00409*.

[38] Lederman, R. R. and Talmon, R. (2018). Learning the geometry of common latent variables using alternating-diffusion. *Applied and Computational Harmonic Analysis*, 44(3):509–536.

[39] Li, C., Shi, J., Zhang, W., Subramanian, A. S., Chang, X., Kamo, N., Hira, M., Hayashi, T., Boeddeker, C., Chen, Z., and Watanabe, S. (2021). ESPnet-SE: End-to-end speech enhancement and separation toolkit designed for ASR integration. In *Proceedings of IEEE Spoken Language Technology Workshop (SLT)*, pages 785–792. IEEE.

[40] Li, J., Lavrukhin, V., Ginsburg, B., Leary, R., Kuchaiev, O., Cohen, J. M., Nguyen, H., and Gadde, R. T. (2019). Jasper: An end-to-end convolutional neural acoustic model. *arXiv preprint arXiv:1904.03288*.

[41] Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. (2019). Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR.

[42] Martinek, R., Kahankova, R., Nazeran, H., Konecny, J., Jezewski, J., Janku, P., Bilik, P., Zidek, J., Nedoma, J., and Fajkus, M. (2017). Non-invasive fetal monitoring: A maternal surface ecg electrode placement-based novel approach for optimization of adaptive filter control parameters using the lms and rls algorithms. *Sensors*, 17(5):1154.

[43] Michaeli, T., Wang, W., and Livescu, K. (2016). Nonparametric canonical correlation analysis. In *International conference on machine learning*, pages 1967–1976. PMLR.

[44] Mor, N., Wolf, L., Polyak, A., and Taigman, Y. (2018). A universal music translation network. *arXiv preprint arXiv:1805.07848*.

[45] Nachmani, E. and Wolf, L. (2019). Unsupervised singing voice conversion. *arXiv preprint arXiv:1904.06590*.

[46] Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572.

[47] Polyak, A., Wolf, L., and Taigman, Y. (2019). Tts skins: Speaker conversion via asr. *arXiv preprint arXiv:1904.08983*.

[48] Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326.

[49] Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319.

[50] Shaham, U. (2018). Batch effect removal via batch-free encoding. *bioRxiv*, page 380816.

[51] Shaham, U. and Lederman, R. R. (2018). Learning by coincidence: Siamese networks and common variable learning. *Pattern Recognition*, 74:52–63.

[52] Sorrenson, P., Rother, C., and Köthe, U. (2020). Disentanglement by nonlinear ica with general incompressible-flow networks (gin). *arXiv preprint arXiv:2001.04872*.

[53] Sprekeler, H., Zito, T., and Wiskott, L. (2014). An extension of slow feature analysis for nonlinear blind source separation. *The Journal of Machine Learning Research*, 15(1):921–947.

[54] Sulas, E., Urru, M., Tumbarello, R., Raffo, L., Sameni, R., and Pani, D. (2021). A non-invasive multimodal foetal ecg–doppler dataset for antenatal cardiology research. *Scientific Data*, 8(1):1–19.

[55] Swarnalath, R. and Prasad, D. (2010). A novel technique for extraction of fecg using multi stage adaptive filtering. *Journal of Applied Sciences*, 10(4):319–324.

[56] Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323.

[57] Veaux, C., Yamagishi, J., MacDonald, K., et al. (2017). Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit. *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*.

[58] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612.

[59] Widrow, B., Glover, J. R., McCool, J. M., Kaunitz, J., Williams, C. S., Hearn, R. H., Zeidler, J. R., Dong, J. E., and Goodlin, R. C. (1975). Adaptive noise cancelling: Principles and applications. *Proceedings of the IEEE*, 63(12):1692–1716.

[60] Wu, S., Shen, Y., Zhou, Z., Lin, L., Zeng, Y., and Gao, X. (2013). Research of fetal ecg extraction using wavelet analysis and adaptive filtering. *Computers in biology and medicine*, 43(10):1622–1627.

[61] Yildirim, O., San Tan, R., and Acharya, U. R. (2018). An efficient compression of ecg signals using deep convolutional autoencoders. *Cognitive Systems Research*, 52:198–211.

[62] Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Limitations are mentioned throughout, and specifically, in Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix C.5.1.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code for reproducing the results in this manuscript is available at `https://github.com/DiSILV`.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] All the technical details for reproducing our experiments are specified in the appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] The standard deviations are presented in Table 1 and Table 2.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] For our experiments we used NVIDIA GTX 1080 GPU. This information appears in the appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We used three publicly available dataset: [57], [54] and [38].
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We included code and samples of converted speech from the voice cloning experiments.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We used only existing publicly available datasets.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] The datasets we used are fully anonymized and do not contain personally identifiable information.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]