SELF-SUPERVISED DISENTANGLED REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Disentanglement has been a central task in representation learning, which involves learning interpretable factors of variation in data. Recent efforts in this direction have been devoted to the identifiability problem of deep latent-variable model with the theory of nonlinear ICA, i.e. the true latent variables can be identified or recovered by the encoder. These identifiability results in nonlinear ICA are essentially based on supervised learning. This work extends these results to the scenario of self-supervised learning. First, we point out that a broad types of augmented data can be generated from a latent model. Based on this, we prove an identifiability theorem similar to the work by (Khemakhem et al., 2020): the latent variables for generating augmented data can be identified with some mild conditions. According to our proposed theory, we perform experiments on synthetic data and EMNIST with GIN (Sorrenson et al., 2020). In our experiments, we find that even the data is only augmented along a few latent variables, more latent variables can be identified, and adding a small noise in data space can stabilize this outcome. Based on this, we augment digit images on EMNIST simply with three affine transformations and then add small Gaussian noise. It is shown that much more interpretable factors of variation can be successfully identified.

1 INTRODUCTION

Disentanglement is to learn representations that each factor corresponds to a single interpretable factor of variation in data sets (Bengio et al., 2013), opening an avenue towards interpretable and more efficient machine learning (Bengio et al., 2013; 2007; Lake et al., 2017; Schmidhuber, 1992; Tschannen et al., 2018). Most previous works focus on tackling this problem by unsupervised learning with InfoGAN (Chen et al., 2016), variational autoencoder (VAE) (Kingma & Welling, 2014) and its variants (Higgins et al., 2017; Burgess et al., 2018; Kim & Mnih, 2018; Chen et al., 2018; Kumar et al., 2018). These works emphasize the necessity of latent variables' independence for disentanglement, by generating synthetic data from a factorial prior or penalizing the total correlation of the learned representations. Besides, some works (Rolinek et al., 2019; Yang et al., 2020) argue that independence of latent variables conditioned on data is also necessary for disentanglement. However, some studies (Hyvarinen & Pajunen, 1999; Locatello et al., 2019) point out that independence of latent variables is not sufficient for identifying them, and thus it is impossible to disentangle interpretable factors of variation in data by simply enforcing their independence.

To identify the latent variables from data is challenging, especially when the latent model is highly nonlinear. A recent breakthrough (Hyvarinen et al., 2019; Khemakhem et al., 2020) in nonlinear Independence Components Analysis (ICA) shows that under relatively mild conditions, it is possible to identify latent variables in a nonlinear latent model, up to a simple transformation. The key requirement is that the latent variables are independent conditioned on an additionally observed variable, and each of them follows a member of exponential family. The work (Sorrenson et al., 2020) further enhances this result, showing that in some special cases, the latent variables can be identified up to an affine transformation and permutation. The above nonlinear ICA theory essentially fits to supervised learning, as an additional observed variable is involved besides raw data.

In this work, we extend the above existing theory to the setting of self-supervised learning. Our key insight is that for a data point, a broad types of augmented data can be generated by manipulating the value of its latent variables. For instance, the augmented data of a digit images from MNIST (LeCun

et al., 1998) can be obtained by manipulating its width or other factors of variation. Therefore, the latent distribution conditioned on data can be viewed as the prior for generating augmented data. Based on this, we assume that the latent variables are independent conditioned on data, and each of them follows a member of exponential family. Then similar with the works by (Hyvarinen et al., 2019; Khemakhem et al., 2020), we prove an identifiability theorem for self-supervised learning. This theory highly coincides with VAEs (Kingma & Welling, 2014) and InfoGAN (Chen et al., 2016), indicating that absolutely unsupervised disentanglement might be possible.

We verify our theory on both synthetic data and EMNIST (Gregory et al., 2017) by using GIN (Sorrenson et al., 2020) as the network architecture, by which each data point and its augmented version are encoded into a factorial Gaussian. The results on synthetic data show that by this way, latent variables can be successfully identified. However, augmenting data along all latent variables is infeasible in many practical scenarios, and is contrary to the purpose of disentanglement. We conjecture that some latent variables not for augmentation are also identifiable, as they also have small changes in augmented data in the learning process. Along this analysis, we further find in experiments that even if we only augment data along a few latent variables, more latent variables can be successfully identified, and adding noise can stabilize this process. According to this empirical result, we augment digit images on EMNIST with three global transformations: translation, rotation and scaling, and add a small Gaussian noise, to more effectively extract the interpretable factors of variation.

Our contributions can be summarized as follows:

1) We extend the recent identifiability results of non-linear ICA (Hyvarinen et al., 2019; Khemakhem et al., 2020) which is closely related to disentanglement learning, to the setting of self-supervised learning. To our best knowledge, this is the first theoretical result for disentangled representation learning by self-supervised learning. It opens up new space for understanding and designing unsupervised disentanglement methods that have not been studied in literature.

2) We show that even using only a few latent variables for generating augmented samples, more latent variables can still be successfully identified, and adding noise can further stabilize this process. The success on this challenging setting has not been reported in existing literature, which indicates our method's generalization ability for complex settings.

3) We further show our method can disentangle interpretable and detailed factors of variation on EMNIST. This has not been achieved in the previous unsupervised methods.

2 Related Works

Most previous works on disentanglement focus on unsupervised learning based on generative adversarial networks (GANs) (Goodfellow et al., 2014) and variational autoencoders (VAEs) (Kingma & Welling, 2014). In the line of GANs, InfoGAN (Chen et al., 2016) generates synthetic data from a factorial prior, and meanwhile recovers the latent variables form the generated data with a factorial Gaussian posterior parameterized by an encoder. As for VAEs, β-VAE (Higgins et al., 2017) penalizes the KL term in VAE's objective to enhance disentanglement. FactorVAE (Kim & Mnih, 2018) penalizes the total correlation of learned representations, i.e. the Kullback-Leibler divergence of the joint distribution and the product of its marginals, by adversarial training. TCVAE (Chen et al., 2018) penalizes the total correlation with its mini-batch estimate. DIP-VAE (Kumar et al., 2018) matches the distribution of the learned representations with the factorial prior. These works are attributed to enhancing the independence of their learned representations (Chen et al., 2018; Locatello et al., 2019). Locatello et al. (2019) point out that independence is not sufficient for disentanglement, as independence cannot ensure the interpretable factors of variation to be identified.

Recent works (Rolinek et al., 2019; Yang et al., 2020) argue that in the unsupervised models above, the factorial posterior is vital for disentanglement, which ensures the factors in the learned representations are independent conditioned on data. Furthermore, these models choose Gaussian, a member of exponential family, as posterior. Hence they fulfill the key condition for nonlinear ICA which will be formally used in this paper: the latent variables are independence conditioned on data, and each of them follows a member of exponential family. Therefore, it is possible to further extend our theory to unsupervised learning and then combine with these models.

Disentangled representation (Bengio et al., 2013) is closely related to ICA. Disentanglement aims at identifying interpretable factors of variation in data, while ICA is to identify the true latent variables. These two goals are equivalent in most cases (Khemakhem et al., 2020; Sorrenson et al., 2020).

Linear ICA focuses on identify or recover latent variables from data which is the mixing of the true latent variables by a linear transformation. The work (Comon, 1994) formulates the problem and gives the first identifiability result, showing that minimizing the total correlation of the learned representations by a linear model leads to identifiability of true latent variables, up to a trivial transformation and permutation. Hyvarinen & Pajunen (1999) prove that this method cannot guarantee identifiability for nonlinear ICA which differs from linear ICA by using nonlinear transformation.

The first identifiability result in nonlinear ICA is given by (Hyvarinen & Morioka, 2016) and (Hyvarinen & Morioka, 2017), and is generalized by (Hyvarinen et al., 2019). Khemakhem et al. (2020) extend the theory into a more general form, and combine it with VAE. Sorrenson et al. (2020) prove that in some cases, the identifiability result can be enhanced up to a trivial transformation and permutation, and meanwhile implement the theoretical model with a volume-preserving Flow-based model called GIN. These works are essentially supervised, as an additionally observed variable is involved. Our theory in this work is an extension of the theory above to self-supervised learning.

This work also bears strong relevance to contrastive self-supervised learning. Typical models like MoCo (He et al., 2019) and SimCLR (Chen et al., 2020) learn representations by maximizing the similarity of each data point with its augmentations, and meanwhile minimizing the similarity with other data to prevent the model to converge as a trivial transformation. Recently a new model called BYOL (Jean-Bastien et al., 2020) argues that minimizing the each data point's similarity with others may be not necessary. Our method can be categorized as contrastive self-supervised learning for disentanglement (not classification)[1], as it encodes each data point and the corresponding augmented data into a factorial Gaussian to improve their similarity. Note that in our method, minimizing each data point's similarity with others is not necessary, as the invertibility of the Flow-based model (Sorrenson et al., 2020) used in our method can prevent the model from converging trivially.

3 THEORY

In this section, we first introduce our self-supervised theory of non-linear ICA, which models the process of generating augmented data with a latent model and gives a preliminary identifiability result. The proof can be found in Appendix A. Then we summarize the further results from (Khemakhem et al., 2020) and (Sorrenson et al., 2020), showing that the latent variables can be identified up to a point-wise affine transformation and permutation.

3.1 OUR SELF-SUPERVISED THEORY OF NONLINEAR ICA

Our key insight is that for a data point, a broad types of augmented data can be generated by a latent model. The generating process consists of three steps: first encode the data point into latent variables, then manipulate their values, and finally decode them to augmented data. Specifically, the progress involves three random variables: a data point $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$, the latent variables $\mathbf{z} \in \mathcal{Z} \subseteq \mathbb{R}^n$ and the augmented data $\mathbf{a} \in \mathcal{A} \subseteq \mathbb{R}^{d'}$, where $\mathcal{X}$, $\mathcal{Z}$ and $\mathcal{A}$ are their supports, respectively. The data point $\mathbf{x}$ can be viewed as the "label" of its augmented data, and hence the latent model above is self-supervised. In addition, the case of $d \neq d'$ is also included in our theory.

The generating progress of latent model above is assumed to be hierarchical: $\mathbf{u} \to \mathbf{z} \to \mathbf{x}$. In other words, let the $\boldsymbol{\theta}$ be the parameters of the latent model:

$$p_{\boldsymbol{\theta}}(\mathbf{a}, \mathbf{z}, \mathbf{x}) = p_{\mathbf{f}}(\mathbf{a}|\mathbf{z}) p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{x}) p(\mathbf{x}) \tag{1}$$

where $p(\mathbf{x})$ is density of the true data. As for $\mathbf{f}$, $\mathbf{T}$ and $\boldsymbol{\lambda}$, they are parameters of the corresponding density functions, and their specific meaning will be explained in the following discussion.

The augmented data $\mathbf{a}$ is assumed as the output of a mixing function $\mathbf{f}$ from $\mathbf{z}$ plus a noise $\boldsymbol{\varepsilon}$: $\mathbf{a} = \mathbf{f}(\mathbf{z}) + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon} \sim p_{\boldsymbol{\varepsilon}}(\boldsymbol{\varepsilon})$. Therefore, the analytic expression of $p_{\mathbf{f}}(\mathbf{a}|\mathbf{z})$ is:

$$p_{\mathbf{f}}(\mathbf{a}|\mathbf{z}) = p_{\boldsymbol{\varepsilon}}(\mathbf{a} - \mathbf{f}(\mathbf{z})) \tag{2}$$

[1]This paper is focused on disentanglement. We leave further applications e.g. classification in future work.

where the mixing $\mathbf{f}$ must be injective, and hence is invertible. The noise $\boldsymbol{\varepsilon}$ makes no difference in our theory, as it can be removed by Fourier transformation (Khemakhem et al., 2020). However, we find adding noise helps identify latent variables even they are not for generating augmented data.

The key assumption of the existing theory is that factors in $\mathbf{z}$ are independent conditioned on $\mathbf{x}$, and their distributions are members of the exponential family with k sufficient statistics:

$$p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{x}) = \prod_{i=1}^{n} p_{\mathbf{T}_i,\boldsymbol{\lambda}_i}(\mathrm{z}_i|\mathbf{x}) = \prod_{i=1}^{n} \frac{Q_i(\mathrm{z}_i)}{Z_i(\lambda_i(\mathbf{x}))} \exp\left[\sum_{j=1}^{k} T_{i,j}(\mathrm{z}_i)\lambda_{i,j}(\mathbf{x})\right] \tag{3}$$

where $\mathbf{T}_i = (T_{i,1}, \cdots, T_{i,k})^\top$ are the sufficient statistics of z_i, and $\boldsymbol{\lambda}_i = (\lambda_{i,1}, \cdots, \lambda_{i,k})^\top$ are the coefficients. Q_i is so-called base measure, often 1, and Z_i is the normalizing factor. $\mathbf{T}(\mathbf{z}) = (\mathbf{T}_1(\mathrm{z}_1)^\top, \cdots, \mathbf{T}_n(\mathrm{z}_n)^\top)^\top = (T_{1,1}(\mathrm{z}_1), \cdots, T_{n,k}(\mathrm{z}_1))^\top$ is the vector of sufficient statistics, and $\boldsymbol{\lambda}(\mathbf{x}) = (\boldsymbol{\lambda}_n(\mathbf{x})^\top, \ldots, \boldsymbol{\lambda}_n(\mathbf{x})^\top)^\top = (\lambda_{1,1}(\mathbf{x}), \cdots, \lambda_{n,k}(\mathbf{x}))^\top$ is the vector of coefficients. k is fixed, representing the dimension of each sufficient statistic.

Many models satisfy the key assumption above, including VAEs (Kingma & Welling, 2014), VIB (Alemi et al., 2017), Flow-based models (Laurent et al., 2015) and etc. These models usually set the posterior as factorial Gaussian, which is a special case in our assumption. Since these models are usually regarded as universal approximators, our assumption is relatively mild.

Based on the model above, following (Hyvarinen et al., 2019; Khemakhem et al., 2020) we can prove the identifiability of $\mathbf{T}(\mathbf{f}^{-1}(\mathbf{a}))$ with some mild conditions:

Theorem 1 *Assume the augmented data of each data point is obtained by the model defined according to equation (1)-(3) with parameters $\boldsymbol{\theta} = (\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda})$, and the following holds:*

(i) $\{\mathbf{a} \in \mathcal{A} | \varphi_{\boldsymbol{\varepsilon}}(\mathbf{a}) = 0\}$ *has measure zero, where $\varphi_{\boldsymbol{\varepsilon}}$ is the characteristic function of $p_{\boldsymbol{\varepsilon}}$.*

(ii) The sufficient statistics $T_{i,j}$ are differentiable almost everywhere, and elements in $\mathbf{T}_i$ are linearly independent on any subset of $\mathcal{Z}$ with non-zero measure, $\forall i \in [n], j \in [k]$.

(iii) $\boldsymbol{\lambda}$ is differentiable, and there exists a data point $\boldsymbol{x}^{(0)}$ such that the Jacobians of $\boldsymbol{\lambda}$ on it $J_{\boldsymbol{\lambda}}(\boldsymbol{x}^{(0)})$ is row full rank.

then if there exists another model defined by Eqs. (1)-(3) with parameter $\tilde{\boldsymbol{\theta}} = (\tilde{\mathbf{f}}, \tilde{\mathbf{T}}, \tilde{\boldsymbol{\lambda}})$. we have:

$$\mathbf{T}(\mathbf{f}^{-1}(\mathbf{a}))) = \boldsymbol{A}\tilde{\mathbf{T}}(\tilde{\mathbf{f}}^{-1}(\mathbf{a})) + \boldsymbol{c}, \forall \mathbf{a} \in \mathcal{A} \tag{4}$$

where $\boldsymbol{A}$ and $\boldsymbol{c}$ are constants, and $\boldsymbol{A}$ is invertible.

In condition *(iii)*, the statement $J_{\boldsymbol{\lambda}}(\boldsymbol{x}^{(0)}) = (\nabla_{\mathbf{x}}\lambda_{1,1}(\boldsymbol{x}^{(0)}), \cdots, \nabla_{\mathbf{x}}\lambda_{n,k}(\boldsymbol{x}^{(0)}))^\top$ is row full rank, is equivalent to that $(\nabla_{\mathbf{x}}\lambda_{1,1}(\boldsymbol{x}^{(0)}), \cdots, \nabla_{\mathbf{x}}\lambda_{n,k}(\boldsymbol{x}^{(0)}))$ are linearly independent. This condition ensures $J_{\boldsymbol{\lambda}}(\boldsymbol{x}^{(0)})^\top$ has left inverse matrix, which is vital in the proof. As the domain of linearly dependent vectors is of zero measure, this condition is almost surely fulfilled for random initialization.

Furthermore, condition *(iii)* reveals a hidden constraint in our theory: the number of true latent variables times k should be not greater than the dimensionality of $\mathbf{x}$, i.e. $n \leq d/k$. Commonly the data point is of high dimensionality, and the number of true latent variables is far more less, hence this constraint is not overly restrictive.

Theoretically speaking, to obtain Eq. (4), a single point $\boldsymbol{x}^{(0)}$ fulfilling condition *(iii)* is sufficient. To see this, consider the members of exponential family have support across the entire latent space $\mathbb{R}^n$. It is obvious that $p(\mathbf{a}|\boldsymbol{x}^{(0)})$ also have support across $\mathcal{A}$. Hence Eq. (4) holds for all $\mathbf{a}$ in $\mathcal{A}$. However, in reality, the augmentations of a single data point are finite, and hence cannot cover $\mathcal{A}$. Therefore, in practice a set of data points fulfilling the condition *(iii)* is necessary to guarantee Eq. (4).

We note that Khemakhem et al. (2020) also introduce a similar version in their appendix. However, it does not ensure $J_{\boldsymbol{\lambda}}(\boldsymbol{x}^{(0)})^\top$ should have left inverse matrix, and is for supervised models. Our theorem is more rigorous and fits for self-supervised learning.

3.2 FURTHER IDENTIFIABILITY

The theory above guarantees the identifiability of $\mathbf{T}(\mathbf{f}^{-1}(\mathbf{a}))$, which can be further enhanced, and finally becomes the identifiability of $\mathbf{z}$ with some conditions. This has been summarized in the following theorems as proposed by (Khemakhem et al., 2020) and (Sorrenson et al., 2020).

Let $\mathbf{z} = \mathbf{f}^{-1}(\mathbf{a})$, $\tilde{\mathbf{z}} = \tilde{\mathbf{f}}^{-1} \circ \mathbf{f}(\mathbf{z})$, then $\mathbf{T}(\mathbf{z}) = (\mathbf{T}_1(\mathrm{z}_1)^\top, \cdots, \mathbf{T}_n(\mathrm{z}_n)^\top)^\top$ is identifiable up to a point-wise affine transform and permutation in some conditions. We quote the original theorem:

Theorem 2 ***(Khemakhem et al., 2020)*** *Assume the hypotheses of Theorem 1 hold, and $k \geq 2$, if:*

(iv) The sufficient statistics $T_{i,j}$ are twice differentiable, $\forall i \in [n], j \in [k]$.

(v) The mixing function $\mathbf{f}$ has all second order cross derivatives

then $\tilde{\mathbf{f}}^{-1} \circ \mathbf{f}$ is a point-wise transformation, and:

$$\mathbf{T}_i(\mathrm{z}_i) = \boldsymbol{A}_i \tilde{\mathbf{T}}_i(\tilde{\mathrm{z}}_i) + \boldsymbol{c}_i \tag{5}$$

The case of $k = 1$ is also reported in (Khemakhem et al., 2020), but the conditions are not so universal due to some special cases. For the sake of analysis we only involve the case of $k \geq 2$.

Sorrenson et al. (2020) prove that when $p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{u})$ and $p_{\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}}}(\tilde{\mathbf{z}}|\mathbf{u})$ are some members of two-parameter exponential family, and $\mathbf{T} = \tilde{\mathbf{T}}$, then $\mathbf{z} = (\mathrm{z}_1, \cdots, \mathrm{z}_n)$ is identifiable up to a point-wise affine transformation and permutation. Here we summarize the case of Gaussian i.e. $\mathbf{T}_i(\mathrm{z}_i) = (\mathrm{z}_i, \mathrm{z}_i^2)$ as for its popularity and rewrite the original theorem as follows:

Theorem 3 ***(Sorrenson et al., 2020)*** *Assume the hypotheses of Theorem 2 hold, if:*

(vi) $\mathbf{T} = \tilde{\mathbf{T}}$ and $\mathbf{T}_i(\mathrm{z}_i) = (\mathrm{z}_i, \mathrm{z}_i^2)$, $\forall i \in [n]$.

then we have:

$$\mathrm{z}_i = a_i \tilde{\mathrm{z}}_i + c_i \tag{6}$$

4 EXPERIMENTS

In this section, we first introduce the our method and optimization, and then introduce experiments on synthetic data and EMNIST, respectively. Experiments on synthetic data verify our theory, and meanwhile show that even though simply a few latent variables are utilized in generating augmented data, more latent variables can be identified. Besides, adding noise to augmented data in data space can stabilizes this outcome. Experiments on EMNIST involve three affine transformations to augment digit images, and show that our method can identify interpretable factors of variation.

4.1 METHOD AND OPTIMIZATION

According to our theory, to identify the true latent variables, it is necessary to encode the data augmented data of each data point into a factorial member of exponential family by an encoder. In experiments, we choose factorial Gaussian as the required factorial member of exponential family, and hence the augmented data of each data point should belong to the same factorial Gaussian in the estimated latent space. For factorial Gaussian, its sufficient statistics of i-th dimension are $\mathbf{T}_i(\mathrm{z}_i) = (\mathrm{z}_i, \mathrm{z}_i^2)$, and the corresponding coefficients are $\boldsymbol{\lambda}_i = (\lambda_{i,1}, \lambda_{i,2}) = (\frac{\mu_i}{\sigma_i^2}, -\frac{1}{2\sigma_i^2})$, where μ_i and σ_i is the mean and variance of the i-th dimension, respectively. Therefore, we can use $(\mu_i(\mathbf{x}), \sigma_i(\mathbf{x}))$ as parameters of the i-th Gaussian directly, rather than $(\lambda_{i,1}(\mathbf{x}), \lambda_{i,2}(\mathbf{x}))$.

As our theory requires an invertible transformation from augmented data to latent space, we choose a volume-preserving flow-based model called GIN (Sorrenson et al., 2020) as encoder. The encoder is parameterized by $\boldsymbol{\theta}$, denoting as E. Note that the input and the output of Flow-based models have the same number of dimensions, and hence the number of output's dimensions will be much larger than the number of true latent variables. This means that the most dimensions in encoder's output are noise, while the others are informative dimensions corresponding to the true latent variables. In

experiments, we enforce all dimensions in estimated latent space to match a Gaussian for augmented data of each data point. By this, the conditions *(ii)* and *(iii)* in our theorem are fulfilled.

Given data points $\mathcal{D} = \left\{\boldsymbol{x}^{(0)}, \cdots, \boldsymbol{x}^{(N)}\right\}$, the first M augmented data points are generated for each data point, i.e. the augmented set is $\mathcal{D}_{aug} = \left\{(\boldsymbol{x}^{(0)}, \{\boldsymbol{a}^{(0,j)}\}_{j=1}^{M}), \cdots, (\boldsymbol{x}^{(N)}, \{\boldsymbol{a}^{(N,j)}\}_{j=1}^{M})\right\}$. Then following (Laurent et al., 2015), we use negative log-likehood of $\mathcal{D}_{aug}$ in the estimated latent space as loss. The Jocobian term equals to 0 as GIN is volume-preserving (Sorrenson et al., 2020):

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x},\mathbf{a})\in\mathcal{D}_{aug}}\left[\frac{1}{n}\sum_{i=1}^{n}\left(\frac{(E(\mathbf{a};\boldsymbol{\theta}) - \mu_i(\mathbf{x};\boldsymbol{\theta}))^2}{2\sigma_i^2(\mathbf{x};\boldsymbol{\theta})} + \log(\sigma_i(\mathbf{x};\boldsymbol{\theta}))\right)\right] \tag{7}$$

In the loss above, for each data point, μ_i and σ_i can be set as the i-th mean and standard deviation of its augmented data in the estimated space (Sorrenson et al., 2020), or estimated by an additional encoder. We find in experiments that these two ways make no difference given enough capacity of the encoder. Thus, we choose the former way for fewer parameters and reduced training time.

4.2 Synthetic Data

Experimental setting. Following (Sorrenson et al., 2020), we synthesize the latent variables, observed data and augmented data by three steps: i) First, two informative latent variables are sampled from a 2-dim mixture of 5 Gaussian, in which the mean of each Gaussian is randomly chosen from a uniform distribution on $[-5, 5]$, and the variance from a uniform distribution on $[0.5, 3]$. From each Gaussian, 2000 points are sampled. Then a 8-dim standard Gaussain noise scaled by 0.01 is concatenated with them. ii) The observed data is generated from the 10-dim latent variables, by a randomly initialized RealNVP (Dinh et al., 2017) with 8 fully connected coupling blocks. iii) For each data point, the augmented data is generated by adding a standard Gaussain noise to its latent variables along the chosen dimensions, and then transforming by the randomly initialized RealNVP. Each data point is augmented as a set with 10 data points.

The training setting is similar with (Sorrenson et al., 2020). The estimating model is a GIN with same setting from (Sorrenson et al., 2020) on artificial data. The training process has 80 epochs, and at each iteration the batch size is 1000 (and hence 10000 augmented data points are used). The optimizer is Adam (Kingma & Ba, 2015) with recommended parameters, and the learning rate is initialized as 10^{-2} and is reduced by a factor of 10 in the last 30 epochs.

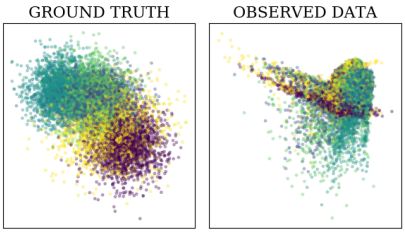
Augmenting along two dimensions. To verify our theory, we first augment each data point along the first two latent dimensions. As shown in Fig. 1 (b), our method can almost perfectly reconstruct the two dimensions, and meanwhile reduce standard deviation of the noise dimensions. This means that our method can identify the true latent variables. Stable results are obtained over 10 trials.

However, augmenting data along all informative dimensions of true latent variables is irrational due to two reasons: i) In most scenes, not all latent variables are known, and hence such augmentation is impossible. ii) As the purpose of disentanglement is exactly to identify true latent variables, if such augmentation is done, then disentanglement is unnecessary. Therefore, we have to consider how to reduce the number of augmented dimensions and meanwhile identify more latent variables.

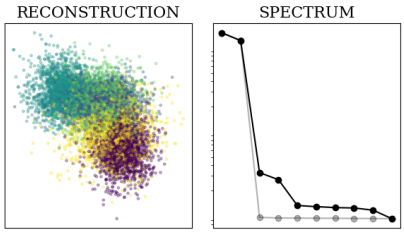
We conjecture that some latent variables not for augmentation are also identifiable, as they also have small changes in augmented data points during the learning process. Note that our method forces each estimated latent dimension to match a Gaussian, and hence the latent variables along all dimensions fulfill the conditions in our theory if they have changes in augmented data. Therefore, it is possible to identify more latent variables in addition with those for augmentation.

Augmenting along one dimension. We find that the two informative dimensions of artificial data can be identified, even though the data points are simply augmented along one of them. As shown in Fig. 1 (c), when data points are augmented along the first latent dimension, the first two latent variables can be successfully identified, but the noise dimensions have higher standard deviations than augmenting along 2 dimensions. This result shows that our method has generalization ability to identify latent variables more than the augmented ones.

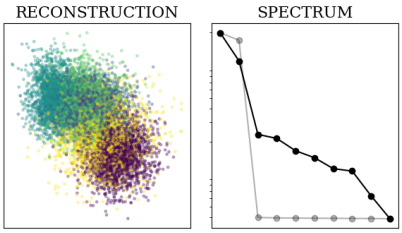
However, this result is unstable. As shown in Fig. 1 (d), the worst reconstruction of latent variables over 10 trials only successfully reconstruct the augmented dimension, and some noise dimensions



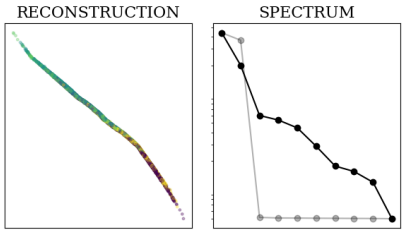
(a) Ground truth and observed data



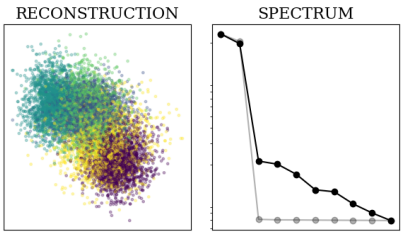
(b) Reconstruction by augmenting along the first two dimensions



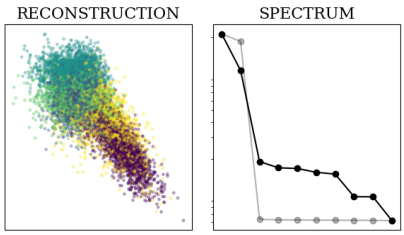
(c) Successful reconstruction by augmenting along the first dimension



(d) Worst reconstruction by augmenting along the first dimension



(e) Successful reconstruction by augmenting along the first dimension and adding noise



(f) Worst reconstruction by augmenting along the first dimension and adding noise

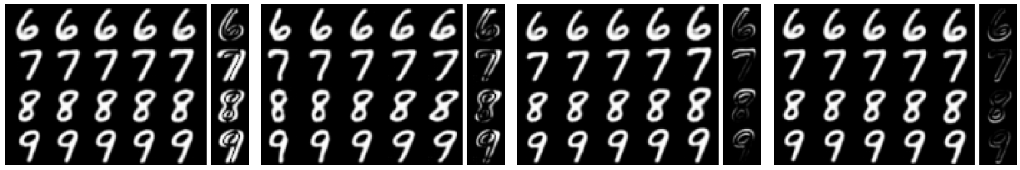
Figure 1: **Ground truth, observed data and reconstruction by different approaches.** (a) ground truth (the first two latent variables) and the observed data (projection to two dimensions). Latent samples from different Gaussian are dyed with different colors. (b) successful reconstruction of the latent variables by augmenting along the first two dimensions. The spectrum is the sorted standard deviations of the estimated variables (in black), while the spectrum of true latent variables is in grey. (c)-(d) successful/worst reconstruction of the first two latent variables by augmenting along the first dimension over 10 trials. (e)-(f) reconstructions by adding a small noise in data space over 10 trials.

have relatively high standard deviations. This is reasonable, as the second dimension is not for augmentations, and hence is ignored in the global optimum.

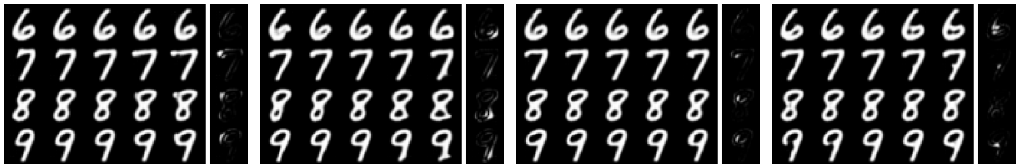
Fortunately, we further find that adding a small noise in data space can stabilize the result above. Specifically, we augment data points along the first latent dimension, and then add a 10-dim standard Gaussian noise scaled by 10^{-2} to augmented data. As shown in Fig. 1 (e), by this way, our method can almost perfectly reconstruct the first two latent variables. The worst reconstruction with noise over 10 trials also much better than the non-noise result, as shown in Fig. 1 (f). In this figure, the first dimension is successfully identified, and the second dimension is also identified with some distortions. These results reveal that adding noise can stabilize the generalization ability.

4.3 EMNIST

EMNIST (Gregory et al., 2017) is an extended version of MNIST, in which we use its Digits training set, including 240,000 images of handwritten digits, as our training set. Each image is augmented by three affine transformations and their combinations: horizontal and vertical translation, rotation and scaling. Their parameters are randomly chosen in certain ranges (see Appendix B). Therefore, each



(a) Horizontal translation (b) Rotation (c) Height of top half (d) Height of bottom half



(e) Width of top half (f) Width of bottom half (g) Size of top cavity (h) Existence of crossbar

Figure 2: **Some interpretable factors of variation on EMNIST disentengled by our method.** Each row is generated by three steps: i) Samples in EMNIST Digits testing set with corresponding label are encoded into estimated latent space by the trained model. ii) Each estimated latent variable is averaged. iii) Manipulate the chosen latent variable on $[-2, 2]$ and then decode the latent variables by the reverse trained model. The rightmost column show the areas affected by the chosen latent variable, computing by the absolute pixel difference on $[-1, 1]$.

image is augmented to 7 images, and we use the augmented images together with the original image as the augmented set. Finally we add a standard Gaussian noise scaled by 0.01 in the augmented set.

The estimating model is a GIN with same setting from (Sorrenson et al., 2020) on EMNIST. The training process runs for 20 epochs, and at each iteration the batch size is 20 (and hence 160 augmented/original images are used). The optimizer is Adam with recommended parameters, and the learning rate is initialized as 3×10^{-2} and is reduced by a factor of 10 in the last 10 epochs.

First, our method can successfully reduce the number of latent dimensions. As shown in Fig. 4 in Appendix C, most latent variables learned by model initialized as identity transformation have high standard deviations. As for the model trained by our method, its spectrum exhibits a sharp knee, and nearly 20 latent variables have high standard deviations.

These 20 latent variables control interpretable and detailed factors of variation on EMNIST. Figures 2 shows part of them, and full set of variables is shown in Appendix C. As shown in Fig. 2 (a)-(b), two variables for augmentations are successfully identified: horizontal translation and rotation. As for the vertical translation and scaling, they are devided into four factors of variation: height of the top/bottom half and width of the top/bottom half, as shown in Fig. 2 (c)-(f).

Note that the height of the top/bottom half are never disentangled by supervised method (Sorrenson et al., 2020). Similarly, size of the top cavity and existence of the crossbar are first disentangled by our method, as shown in Fig. 2 (g)-(h). These are realistic factors of variation on EMNSIT. In contrast, there exists some interpretable factors of variation on EMNIST not discovered by our method, like line thickness. These results reveal that our method disentangles factors of variation from the dataset with a way different from the supervised method (Sorrenson et al., 2020).

5 Conclusion and Outlook

We have extended the identifiability results of non-linear ICA to self-supervised learning, proving that under relatively mild conditions, the latent variables for augmentation are identifiable. Based on this, we apply this theory to identify latent variables from data for disentanglement. We found in experiments that our method has certain generalization ability to identify more latent variables than those for augmentation, and adding a small noise in data space can further stabilize this process. Experiments on EMNIST also show that our method can disentangle interpretable and detailed factors of variation. Our theory is very promising to be further extended to unsupervised learning, as the assumptions in our theory has been highly close to the structures of VAEs and InfoGAN. Moreover, how to explain the generalization ability of our method theoretically is still an open problem.

REFERENCES

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *International Conference on Learning Representations*, 2017.

Yoshua Bengio, Yann LeCun, et al. Scaling learning algorithms towards ai. *Large-scale Kernel Machines*, 34(5):1–41, 2007.

Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1798–1828, 2013.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *arXiv: Machine Learning*, 2018.

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

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E Hinton. A simple framework for contrastive learning of visual representations. *arXiv: 2002.05709*, 2020.

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

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

Laurent Dinh, Jascha Sohldickstein, and Samy Bengio. Density estimation using real nvp. *International Conference on Learning Representations*, 2017.

Ian J Goodfellow, Jean Pougetabadie, Mehdi Mirza, Bing Xu, David Wardefarley, Sherjil Ozair, Aaron C Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

Cohen Gregory, Afshar Saeed, Tapson Jonathan, , and van Schaik Andre. Emnist: an extension of mnist to handwritten letters. *arXiv:1702.05373*, 2017.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv: 1911.05722*, 2019.

Irina Higgins, Loic Matthey, Arka Pal, Christopher P Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-vae: Learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations*, 2017.

Aapo Hyvarinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. *Advances in Neural Information Processing Systems*, pp. 3772–3780, 2016.

Aapo Hyvarinen and Hiroshi Morioka. Nonlinear ica of temporally dependent stationary sources. *International Conference on Artificial Intelligence and Statistics*, pp. 460–469, 2017.

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

Aapo Hyvarinen, Hiroaki Sasaki, and Richard E Turner. Nonlinear ica using auxiliary variables and generalized contrastive learning. *International Conference on Artificial Intelligence and Statistics*, pp. 859–868, 2019.

Grill Jean-Bastien, Strub Florian, Altché Florent, Tallec Corentin, H. Richemond Pierre, Buchatskaya Elena, Doersch Carl, Pires Bernardo Avila, Guo Zhaohan Daniel, Azar Mohammad Gheshlaghi, Piot Bilal, Kavukcuoglu Koray, Munos Rémi, and Valko Michal. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv: 2006.07733*, 2020.

Ilyes Khemakhem, Diederik P Kingma, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. *International Conference on Artificial Intelligence and Statistics*, 2020.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *International Conference on Machine Learning*, pp. 2649–2658, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *International Conference on Learning Representations*, 2014.

Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. *International Conference on Learning Representations*, 2018.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017.

Dinh Laurent, Krueger David, and Bengio Yoshua. Nice: Non-linear independent components estimation. *arXiv: 1410.8516*, 2015.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Francesco Locatello, Stefan Bauer, Mario Lucic, Sylvain Gelly, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *International Conference on Machine Learning*, pp. 4114–4124, 2019.

Michal Rolinek, Dominik Zietlow, and Georg Martius. Variational autoencoders pursue pca directions (by accident). *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12406–12415, 2019.

Jürgen Schmidhuber. Learning factorial codes by predictability minimization. *Neural Computation*, 4(6):863–879, 1992.

Peter Sorrenson, Carsten Rother, and Ullrich Kothe. Disentanglement by nonlinear ica with general incompressible-flow networks (gin). *International Conference on Learning Representations*, 2020.

Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*, 2018.

Xiaojiang Yang, Wendong Bi, Yitong Sun, Yu Cheng, and Junchi Yan. Towards better understanding of disentangled representations via mutual information. *arXiv: 1911.10922*, 2020.

APPENDIX

A PROOF OF THEOREM 1

Theorem 1 *Assume the augmented data of each data point is obtained by the model defined according to equation (1)-(3) with parameters $\boldsymbol{\theta} = (\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda})$, and the following holds:*

(i) $\{\mathbf{a} \in \mathcal{A} | \varphi_{\boldsymbol{\varepsilon}}(\mathbf{a}) = 0\}$ *has measure zero, where $\varphi_{\boldsymbol{\varepsilon}}$ is the characteristic function of $p_{\boldsymbol{\varepsilon}}$.*

(ii) The sufficient statistics $T_{i,j}$ are differentiable almost everywhere, and elements in $\mathbf{T}_i$ are linearly independent on any subset of $\mathcal{Z}$ with non-zero measure, $\forall i \in [n], j \in [k]$.

(iii) $\boldsymbol{\lambda}$ is differentiable, and there exists a data point $\boldsymbol{x}^{(0)}$ such that the Jacobians of $\boldsymbol{\lambda}$ on it $J_{\boldsymbol{\lambda}}(\boldsymbol{x}^{(0)})$ is row full rank.

then if there exists another model defined by Eqs. (1)-(3) with parameter $\tilde{\boldsymbol{\theta}} = (\tilde{\mathbf{f}}, \tilde{\mathbf{T}}, \tilde{\boldsymbol{\lambda}})$. we have:

$$\mathbf{T}(\mathbf{f}^{-1}(\mathbf{a}))) = \boldsymbol{A}\tilde{\mathbf{T}}(\tilde{\mathbf{f}}^{-1}(\mathbf{a})) + \boldsymbol{c}, \forall \mathbf{a} \in \mathcal{A} \tag{8}$$

where $\boldsymbol{A}$ and $\boldsymbol{c}$ are constants, and $\boldsymbol{A}$ is invertible.

proof. The proof consists of three steps, in which the first step and the third step are quoted from (Khemakhem et al., 2020).

Step I. Suppose there exists two sets of parameters $\boldsymbol{\theta} = (\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda})$ and $\tilde{\boldsymbol{\theta}} = (\tilde{\mathbf{f}}, \tilde{\mathbf{T}}, \tilde{\boldsymbol{\lambda}})$ such that $p_{\mathbf{f},\mathbf{T},\boldsymbol{\lambda}}(\mathbf{a}|\mathbf{x}) = p_{\tilde{\mathbf{f}},\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}}}(\mathbf{a}|\mathbf{x}), \forall (\mathbf{a}, \mathbf{x}) \in (\mathcal{A}, \mathcal{X})$. Then:

$$\int_{\mathcal{Z}} p_{\mathbf{f}}(\mathbf{a}|\mathbf{z}) p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{x}) d\mathbf{z} = \int_{\mathcal{Z}} p_{\tilde{\mathbf{f}}}(\mathbf{a}|\mathbf{z}) p_{\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}}}(\mathbf{z}|\mathbf{x}) d\mathbf{z} \tag{9}$$

Substitute the expression of $p_{\mathbf{f}}(\mathbf{a}|\mathbf{z})$:

$$\int_{\mathcal{Z}} p_{\varepsilon}(\mathbf{a} - \mathbf{f}(\mathbf{z})) p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{x}) d\mathbf{z} = \int_{\mathcal{Z}} p_{\varepsilon}(\mathbf{a} - \tilde{\mathbf{f}}(\mathbf{z})) p_{\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}}}(\mathbf{z}|\mathbf{x}) d\mathbf{z} \tag{10}$$

Transform $\mathbf{z}$ into $\mathcal{A}$, denoted as $\bar{\mathbf{a}}$:

$$\int_{\mathcal{A}} p_{\varepsilon}(\mathbf{a} - \bar{\mathbf{a}}) p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{f}^{-1}(\bar{\mathbf{a}})|\mathbf{x}) \mathrm{vol} J_{\mathbf{f}^{-1}}(\bar{\mathbf{a}}) d\bar{\mathbf{a}} = \int_{\mathcal{A}} p_{\varepsilon}(\mathbf{a} - \bar{\mathbf{a}}) p_{\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}}}(\tilde{\mathbf{f}}^{-1}(\bar{\mathbf{a}})|\mathbf{x}) J_{\tilde{\mathbf{f}}^{-1}}(\bar{\mathbf{a}}) d\bar{\mathbf{a}} \tag{11}$$

This is a convolution on $\mathbf{a}$, and the kernal is $p_{\varepsilon}(\mathbf{a})$. Use Fourier transformation, and the transformed $\mathbf{a}$ is denoted as w:

$$F[p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{f}^{-1}(\mathbf{a})|\mathbf{x}) \mathrm{vol} J_{\mathbf{f}^{-1}}(\mathbf{a})] \varphi_{\varepsilon}(w) = F[p_{\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}}}(\tilde{\mathbf{f}}^{-1}(\mathbf{a})|\mathbf{x}) J_{\tilde{\mathbf{f}}^{-1}}(\mathbf{a})] \varphi_{\varepsilon}(w) \tag{12}$$

According to condition *(i)*, $\varphi_{\varepsilon}(w) \neq 0$ almost every where. Hence we have:

$$F[p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{f}^{-1}(\mathbf{a})|\mathbf{x}) \mathrm{vol} J_{\mathbf{f}^{-1}}(\mathbf{a})] = F[p_{\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}}}(\tilde{\mathbf{f}}^{-1}(\mathbf{a})|\mathbf{x}) J_{\tilde{\mathbf{f}}^{-1}}(\mathbf{a})] \tag{13}$$

Use the inverse Fourier transformation, we finally obtain:

$$p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{f}^{-1}(\mathbf{a})|\mathbf{x}) \mathrm{vol} J_{\mathbf{f}^{-1}}(\mathbf{a}) = p_{\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}}}(\tilde{\mathbf{f}}^{-1}(\mathbf{a})|\mathbf{x}) J_{\tilde{\mathbf{f}}^{-1}}(\mathbf{a}) \tag{14}$$

This result shows that noise fulfilled condition *(i)* makes no difference theoretically.

Step II. Substitute the expression of $p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{f}^{-1}(\mathbf{a})|\mathbf{x})$ in Eq. (14), and take the log on the both sides:

$$\begin{aligned} &\log \mathrm{vol} J_{\mathbf{f}^{-1}}(\mathbf{a}) + \sum_{i=1}^{n} \left(\log Q_i(f_i^{-1}(\mathbf{a})) - \log Z_i(\boldsymbol{\lambda}_i(\mathbf{x})) + \mathbf{T}_i(f_i^{-1}(\mathbf{a}))^{\top} \boldsymbol{\lambda}_i(\mathbf{x}) \right) \\ =& \log \mathrm{vol} J_{\tilde{\mathbf{f}}^{-1}}(\mathbf{a}) + \sum_{i=1}^{n} \left(\log \tilde{Q}_i(\tilde{f}_i^{-1}(\mathbf{a})) - \log \tilde{Z}_i(\tilde{\boldsymbol{\lambda}}_i(\mathbf{x})) + \tilde{\mathbf{T}}_i(\tilde{f}_i^{-1}(\mathbf{a}))^{\top} \tilde{\boldsymbol{\lambda}}_i(\mathbf{x}) \right) \end{aligned} \tag{15}$$

Take the first derivative with respect to $\mathbf{x}$:

$$J_{\boldsymbol{\lambda}}(\mathbf{x})^{\top} \mathbf{T}(\mathbf{f}^{-1}(\mathbf{a})) - \sum_{i=1}^{n} J_{\boldsymbol{\lambda}_i}(\mathbf{x})^{\top} \nabla_{\boldsymbol{\lambda}_i} \log Z_i(\boldsymbol{\lambda}_i) = J_{\tilde{\boldsymbol{\lambda}}}(\mathbf{x})^{\top} \tilde{\mathbf{T}}(\tilde{\mathbf{f}}^{-1}(\mathbf{a})) - \sum_{i=1}^{n} J_{\tilde{\boldsymbol{\lambda}}_i}(\mathbf{x})^{\top} \nabla_{\tilde{\boldsymbol{\lambda}}_i} \log \tilde{Z}_i(\tilde{\boldsymbol{\lambda}}_i) \tag{16}$$

Table 1: Ranges of three transformations.

Transformation	Range
translation	$[-0.1, 0.1] \times [-0.1, 0.1]$
rotation	$[0, 5°]$
scaling	$[0.8, 1.25]$

Figure 3: **Five samples and their augmented images by the three transformations.** The first row shows the original images. The next three rows show the images augmented by horizontal and vertical translation, rotation and scaling, respectively.

According to condition *(iii)*, let $\mathbf{x} = \mathbf{x}^{(0)}$, then $J_{\boldsymbol{\lambda}}(\boldsymbol{x}^{(0)})^\top$ is row full rank, and hence has left pseudo inverse matrix: $\left(J_{\boldsymbol{\lambda}}(\boldsymbol{x}^{(0)})J_{\boldsymbol{\lambda}}(\boldsymbol{x}^{(0)})^\top\right)^{-1}J_{\boldsymbol{\lambda}}(\boldsymbol{x}^{(0)})$. Multiply the both sides by the left pseudo inverse matrix, we have:

$$\mathbf{T}(\mathbf{f}^{-1}(\mathbf{a})) = \boldsymbol{A}\tilde{\mathbf{T}}(\tilde{\mathbf{f}}^{-1}(\mathbf{a})) + \boldsymbol{c} \tag{17}$$

where:

$$\begin{aligned}
\boldsymbol{A} &= \left(J_{\boldsymbol{\lambda}}(\boldsymbol{x}^{(0)})J_{\boldsymbol{\lambda}}(\boldsymbol{x}^{(0)})^\top\right)^{-1}J_{\boldsymbol{\lambda}}(\boldsymbol{x}^{(0)})J_{\tilde{\boldsymbol{\lambda}}}(\boldsymbol{x}^{0})^\top \\
\boldsymbol{c} &= \left(J_{\boldsymbol{\lambda}}(\boldsymbol{x}^{(0)})J_{\boldsymbol{\lambda}}(\boldsymbol{x}^{(0)})^\top\right)^{-1}J_{\boldsymbol{\lambda}}(\boldsymbol{x}^{(0)})\sum_{i=1}^{n}\left(J_{\boldsymbol{\lambda}_i}(\boldsymbol{x}^{(0)})^\top\nabla_{\boldsymbol{\lambda}_i}\log Z_i(\boldsymbol{\lambda}_i) - J_{\tilde{\boldsymbol{\lambda}}_i}(\boldsymbol{x}^{(0)})^\top\nabla_{\tilde{\boldsymbol{\lambda}}_i}\log\tilde{Z}_i(\tilde{\boldsymbol{\lambda}}_i)\right)
\end{aligned} \tag{18}$$

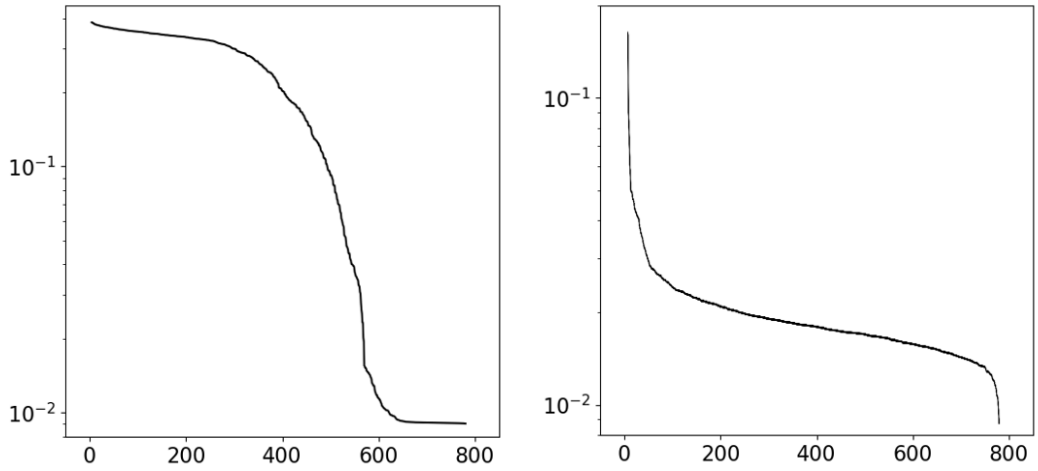
Note that $p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{x})$ has support cross the entire latent space $\mathbb{R}^n$, and hence $p_{\mathbf{f},\mathbf{T},\boldsymbol{\lambda}}(\mathbf{a}|\boldsymbol{x}^0)$ has support cross $\mathcal{A}$. As a result, Eq. 17 holds for any $\mathbf{a} \in \mathcal{A}$. This means that $\boldsymbol{A}$ and $\boldsymbol{c}$ are independent of $\mathbf{x}$.

Step III. In this step, we prove the invertibility of $\boldsymbol{A}$. Let $\mathbf{z} = \mathbf{f}^{-1}(\mathbf{a})$, then Eq. 17 can be written as: $\mathbf{T}(\mathbf{z}) = \boldsymbol{A}\tilde{\mathbf{T}}(\tilde{\mathbf{f}}^{-1} \circ \mathbf{f}(\mathbf{z})) + \boldsymbol{c}$. Take the first derivative with respect to $\mathbf{z}$, then: $J_{\mathbf{T}}(\mathbf{z}) = \boldsymbol{A}J_{\mathbf{T}\circ\mathbf{f}^{-1}\circ\mathbf{f}}(\mathbf{z})$. According to the Lemma by (Khemakhem et al., 2020), by condition *(ii)*, there exits a k distinct latent samples $\{\boldsymbol{z}^{(1)}, \cdots, \boldsymbol{z}^{(k)}\}$ such that is $(J_{\mathbf{T}}(\boldsymbol{z}^{(1)}), \cdots, J_{\mathbf{T}}(\boldsymbol{z}^{(k)}))$ is invertible. Note that $(J_{\mathbf{T}}(\boldsymbol{z}^{(1)}), \cdots, J_{\mathbf{T}}(\boldsymbol{z}^{(k)})) = \boldsymbol{A}(J_{\mathbf{T}\circ\mathbf{f}^{-1}\circ\mathbf{f}}(\boldsymbol{z}^{(1)}), \cdots, J_{\mathbf{T}\circ\mathbf{f}^{-1}\circ\mathbf{f}}(\boldsymbol{z}^{(k)}))$, hence $\boldsymbol{A}$ is invertible.

B AUGMENTATIONS ON EMNIST

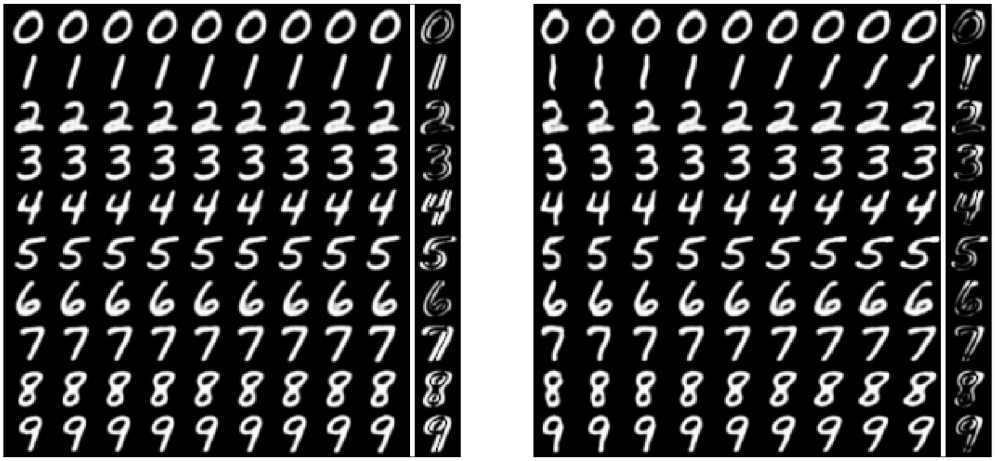
The augmentations used on EMNIST in this work includes three affine transformations: horizontal and vertical translation, rotation and scaling. Their parameters are randomly chosen from a uniform distribution on certain ranges for each data point. The ranges are shown in Table B. Five images augmented by these transformations are shown in Fig. 3. Note that in experiments, we use the three transformations and their combinations, totally 7 types, for augmentations.

C FIGURES FROM THE EMNIST EXPERIMENTS



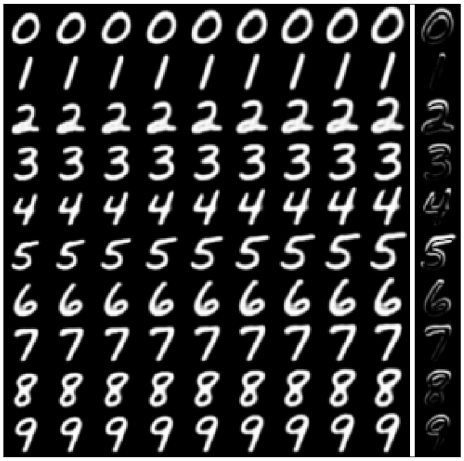
(a) Spectrum from initialized model (b) Spectrum from trained model

Figure 4: Spectrum of sorted standard deviations from model initialized by identity transformation and the trained model. X-axis is sorted latent dimensions; y-axis is standard deviations in log scale.

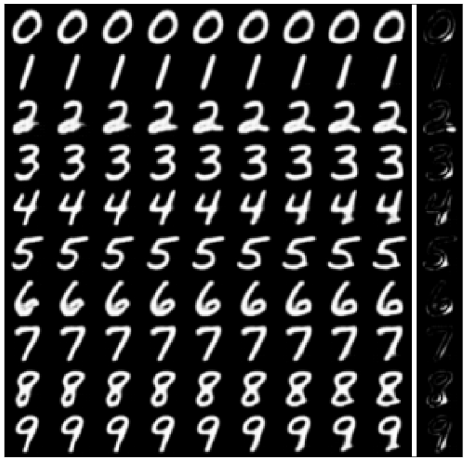


(a) Variable 1: horizontal translation (b) Variable 2: rotation

Figure 5: Most significant latent variables 1-2. Each row is generated by three steps: i) Samples in EMNIST Digits testing set with corresponding label are encoded into estimated latent space by the trained model. ii) Each estimated latent variable is averaged. iii) Manipulate the chosen latent variable on $[-2, 2]$ and then decode the latent variables by the reverse trained model. The rightmost column show the areas affected by the chosen latent variable, computing by the absolute pixel difference on $[-1, 1]$.

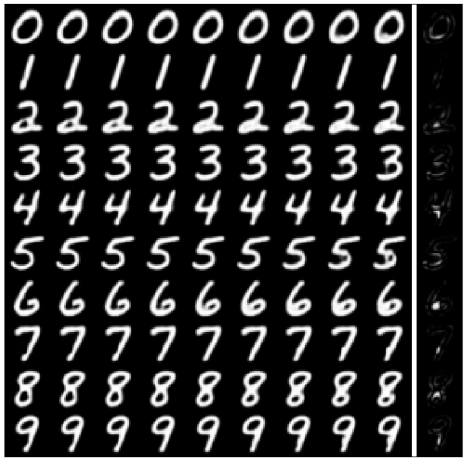


(a) Variable 3: height of the top half



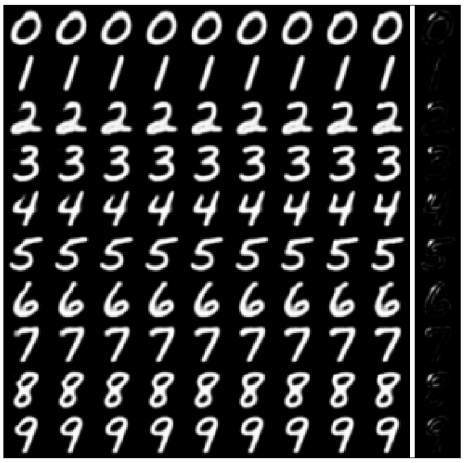
(b) Variable 4: width of the bottom half by the bottom right corner

(c) Variable 5: height of the bottom half

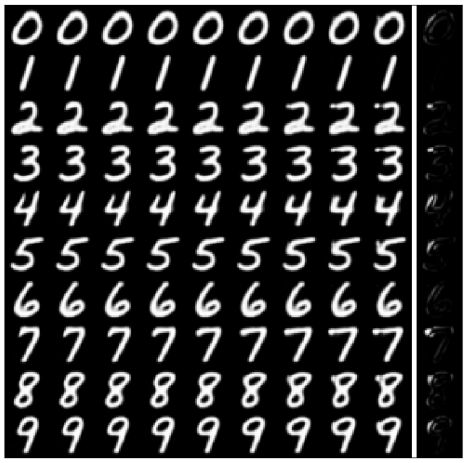


(d) Variable 6: black space of the lower half

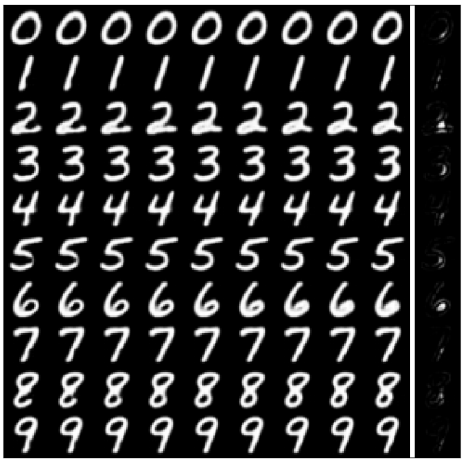
Figure 6: Most significant latent variables 3-6. Each row is generated by three steps: i) Samples in EMNIST Digits testing set with corresponding label are encoded into estimated latent space by the trained model. ii) Each estimated latent variable is averaged. iii) Manipulate the chosen latent variable on $[-2, 2]$ and then decode the latent variables by the reverse trained model. The rightmost column show the areas affected by the chosen latent variable, computing by the absolute pixel difference on $[-1, 1]$.



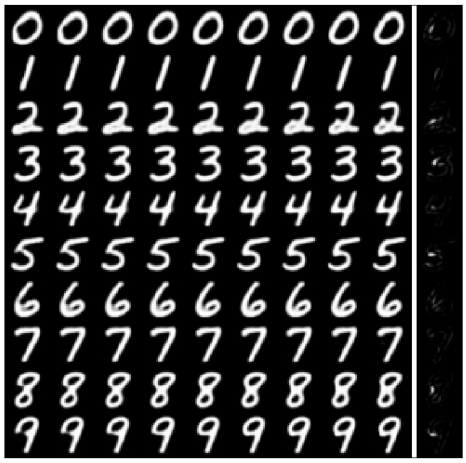
(a) Variable 7: width of the top half by the top left corner



(b) Variable 8: width of the top half by the lower top left corner

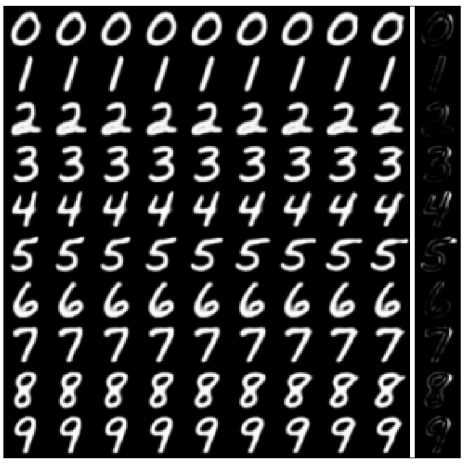


(c) Variable 9: openess of the bottom half

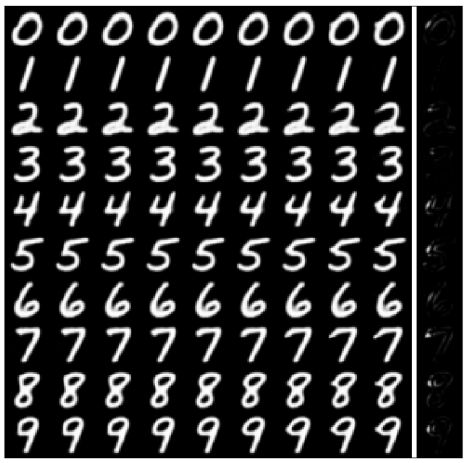


(d) Variable 10: openess of the top half

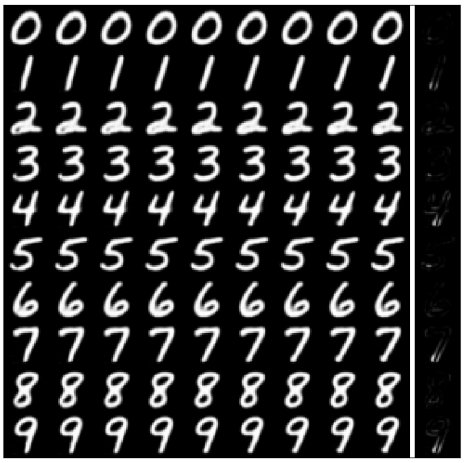
Figure 7: Most significant latent variables 7-10. Each row is generated by three steps: i) Samples in EMNIST Digits testing set with corresponding label are encoded into estimated latent space by the trained model. ii) Each estimated latent variable is averaged. iii) Manipulate the chosen latent variable on $[-2, 2]$ and then decode the latent variables by the reverse trained model. The rightmost column show the areas affected by the chosen latent variable, computing by the absolute pixel difference on $[-1, 1]$.



(a) Variable 11: top right corner



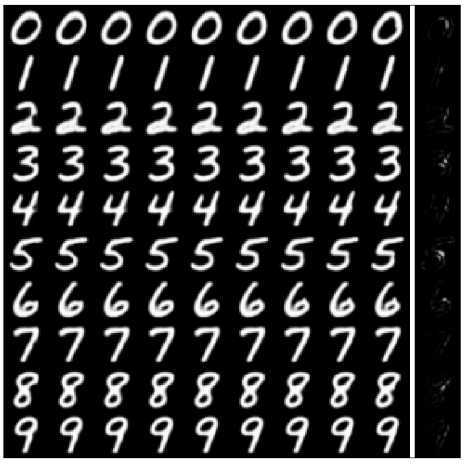
(b) Variable 12: lower top left coner



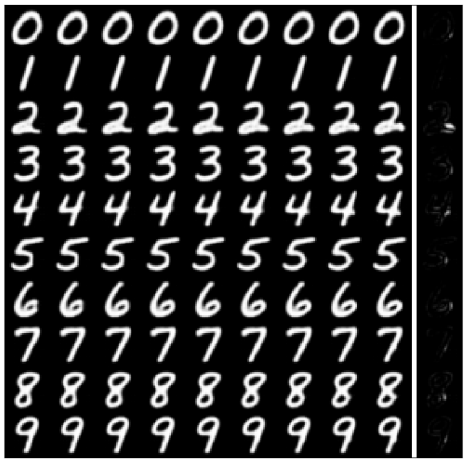
(c) Variable 13: bend of vertical bar in 1, 4, 7, 9

(d) Variable 14: extension of right corner

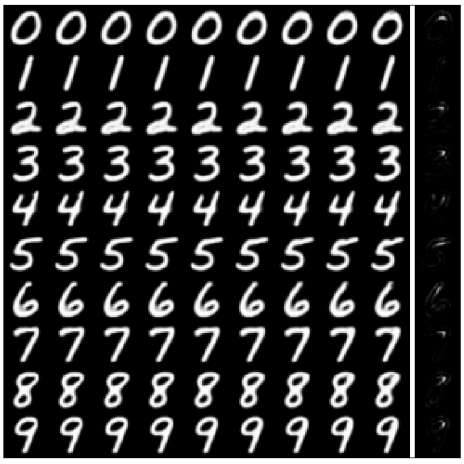
Figure 8: Most significant latent variables 11-14. Each row is generated by three steps: i) Samples in EMNIST Digits testing set with corresponding label are encoded into estimated latent space by the trained model. ii) Each estimated latent variable is averaged. iii) Manipulate the chosen latent variable on $[-2, 2]$ and then decode the latent variables by the reverse trained model. The rightmost column show the areas affected by the chosen latent variable, computing by the absolute pixel difference on $[-1, 1]$.



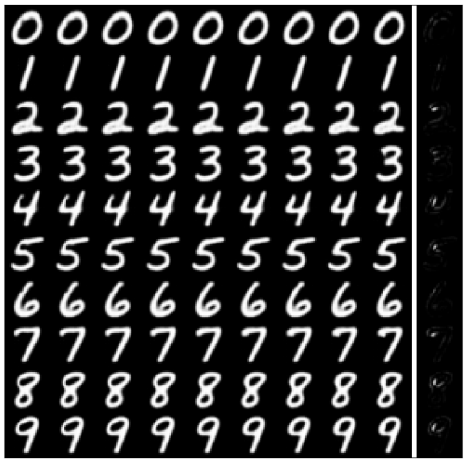
(a) Variable 15: extension of higher right corner



(b) Variable 16: extension of lower right corner

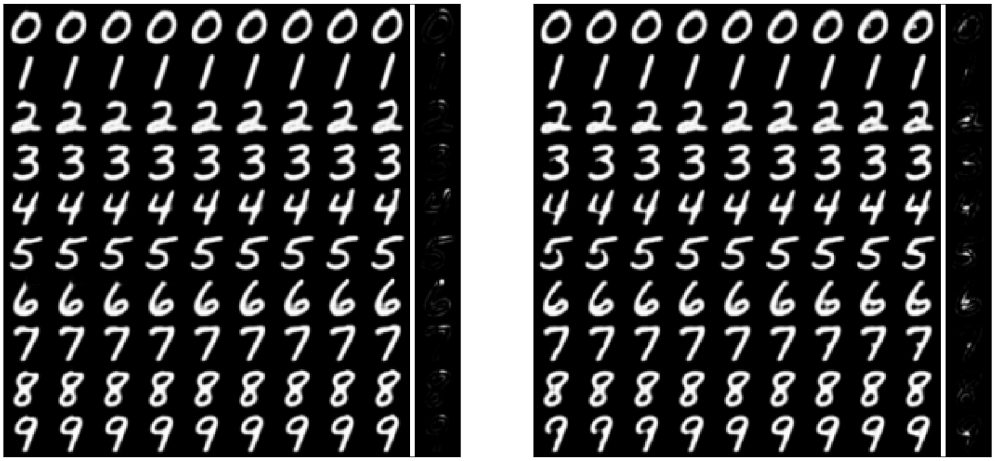


(c) Variable 17: size of top capity by lower top right corner



(d) Variable 18: size of top capity by lower top left corner

Figure 9: Most significant latent variables 15-18. Each row is generated by three steps: i) Samples in EMNIST Digits testing set with corresponding label are encoded into estimated latent space by the trained model. ii) Each estimated latent variable is averaged. iii) Manipulate the chosen latent variable on $[-2, 2]$ and then decode the latent variables by the reverse trained model. The rightmost column show the areas affected by the chosen latent variable, computing by the absolute pixel difference on $[-1, 1]$.



(a) Variable 19: bend of the top horizontal bar

(b) Variable 20: existence of the crossbar

Figure 10: Most significant latent variables 19-20. Each row is generated by three steps: i) Samples in EMNIST Digits testing set with corresponding label are encoded into estimated latent space by the trained model. ii) Each estimated latent variable is averaged. iii) Manipulate the chosen latent variable on $[-2, 2]$ and then decode the latent variables by the reverse trained model. The rightmost column show the areas affected by the chosen latent variable, computing by the absolute pixel difference on $[-1, 1]$.