# Learning Switchable Representation with Masked Decoding and Sparse Encoding

**Anonymous Authors**[1]

## Abstract

In this study, we explore the unsupervised learning based on private/shared factor decomposition, which decomposes the latent space into *private* factors that vary only in a specific domain and the *shared* factors that vary in all domains. We study when/how we can force the model to respect the true private/shared factor decomposition that underlies the dataset. We show that, when we train a masked decoder and an encoder with sparseness regularization in the latent space, we can identify the correct private/shared decomposition up to mixing within each component. We empirically confirm this result and study the efficacy of this training strategy as a representation learning method.

## 1. Introduction

Finding a few common factors that consistently explain different phenomena is a fundamental challenge for unsupervised learning. For example, consider two datasets of car images. One consists of synthetic images of 3D objects generated with realistic optical rules (physically based rendering), and the another consists of real images. Their appearance should be different, for example, in the details of images and the texture of the objects. However, their physical properties, such as the positioning of objects and the way the image changes with respect to camera positions, shall all be the same between the synthetic and real datasets. Many of these physical properties explain the structure within each dataset; abstractly, they correspond to the axis $A$, $B$ and $C$ in the visualization in Figure 1.

Unfortunately, however, the mechanism of finding such commonalities has not been fully established. For the above datasets, let us consider applying principal component analysis (PCA). PCA extracts the subspace in which the variation of the data in the observation space is maximized. If the variation across the two datasets is larger than the variation within each dataset, PCA would first extract the axis that is specialized in explaining the difference between the two datasets. However, this might cause some problems because *this* principal axis might be highly correlated to important factors of variation in each dataset (A, B, C in Figure 1). Thus, simply removing the factor of the largest difference might lose the important structural information within each dataset. Also, because of this entanglement between the first principal component and A, B, and C, it will not be possible to align the rest of the principal components to these structural factors of variations.

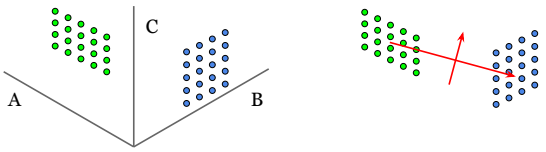



*Figure 1.* Example of unsupervised representation learning on two domains. Left: observations from the domain 1 (green) depend on latent factors $A$ and $C$ while the observations from the domain 2 (blue) depend on $B$ and $C$. Right: The principal vectors of PCA do not align with $A$, $B$, and $C$.

To obtain the desired latent structure, we have to choose an appropriate inductive bias (Locatello et al., 2019). PCA does not work well in the previous example because the objective of PCA is based on the belief that the axis with the largest variance corresponds to the most important structural factor. One way to resolve this problem is to introduce the shared-private decomposition of latent space (Bousmalis et al., 2016; Liu et al., 2017; Cao et al., 2018; Peng et al., 2019; Chattopadhyay et al., 2020; Bui et al., 2021). In this framework, the domain-private factors consist of the factors that vary only in one individual domain, whereas the shared-domain factors consist of the factors that vary in all domains. By introducing such a structure in the generating process, this strategy imposes the inductive bias that the set of data-describing structural factors includes the features shared by all domains (e.g., $C$ in Figure 1.) However, even if we introduce such an inductive bias, it is not entirely clear if the trained model would identify the true private/shared

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

structure in the dataset.

In this work, we investigate when we can use the private/shared inductive bias to extract the true private/shared factors so that we can use them for improvement on representation learning and extrapolation tasks. We provide an empirical and basic theoretical result indicating that, when the decoder uses a domain-specific mask function, we can align the feature space to the private-vs-shared decomposition by encouraging the sparsity in the latent space. We provide an empirical and basic theoretical result indicating that we can align the feature space to the private-vs-shared decomposition by encouraging the sparsity in the latent space.

## 2. Method

In this paper, we consider the multi-domain setting in which the domain membership of each datum is known. Therefore we assume that each observation is a pair $(\mathbf{x}, k)$, where $\mathbf{x} \in \mathbb{R}^d$ is the raw observation and $k \in \{1, \ldots, L\}$ is the domain label from which $\mathbf{x}$ was drawn. Our goal is to recover the latent factors $\mathbf{z} \in \mathbb{R}^m$ when the true generating process has private factors and shared factors.

We elaborate the assumed generating process below. In our setting, the latent factors are separated into $L+1$ parts as $\mathbf{z} = [\mathbf{z}_1, \ldots, \mathbf{z}_L, \mathbf{z}_{\text{shared}}]$ where each $\mathbf{z}_k \in \mathbb{R}^{m_k}$ represents the private factor of the domain $k$ and $\mathbf{z}_{\text{shared}} \in \mathbb{R}^{m_{\text{shared}}}$ represents the shared factor so that $m = m_{\text{shared}} + \sum_k m_k$. For brevity, let $\mathcal{I}_k$ represent the set of coordinate indices in $\mathbb{R}^m$ corresponding to the $k$th domain, and let $\mathcal{I}_{\text{shared}}$ represent the set of coordinate indices for the shared factors. We prepare such a structure in the latent space so that the features that will be trained by the encoder will not entangle the domain-specific features.

The generative process of each observation in the domain $k$ is then written as

$$p(\mathbf{x} \mid \mathbf{z}, k) = p(\mathbf{x} \mid \mathbf{z}_k, \mathbf{z}_{\text{shared}}). \tag{1}$$

Based on this model, we train an autoencoder to seek the underlying structure of the observation that aligns with our decomposition. Let $f : \mathbb{R}^d \to \mathbb{R}^m$ be the encoder and $g : \mathbb{R}^m \to \mathbb{R}^d$ be the decoder. We define the objective function as

$$\frac{1}{2}\|\mathbf{x} - g(\mathbf{u}_k \odot \mathbf{z})\|^2 + \beta r(\mathbf{z}) \quad \text{with} \quad \mathbf{z} = f(\mathbf{x}), \tag{2}$$

where $\odot$ denotes the element-wise multiplication, $r(\cdot)$ is a sparsity regularizer for $\mathbf{z}$, and $\beta \geq 0$ controls the sparsity. Here, $\mathbf{u}_k \in \{0,1\}^m$ is the binary mask such that $\mathbf{u}_{ki} = 1$ whenever $i \in \mathcal{I}_k \cup \mathcal{I}_{\text{shared}}$. The masks $\mathbf{u}_k$ are not trainable parameters in our objective. If there are two domains to be studied, then the masks shall therefore look

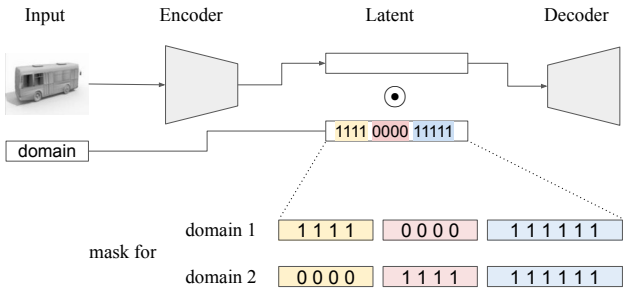



*Figure 2.* Overview of the proposed method.

like $\mathbf{u}_1 = [\mathbf{1}, \mathbf{0}, \mathbf{1}]$ and $\mathbf{u}_2 = [\mathbf{0}, \mathbf{1}, \mathbf{1}]$ where $\mathbf{1}$ indicates the all-one vector with appropriate dimension (Figure 2). In other words, the mask $\mathbf{u}_k$ is used to switch between the different domains. For example, if the images are colored in the domain $k$ and monocolor in all other domains, then $\mathbf{z}_k$ may contain color features. Mask $\mathbf{u}_k$ is an integral part of the decoder because $z_k$ won't be used in decoding $x$ from $z$ when $x$ is in the domain $k' \neq k$.

We note that the mask is a structure that is included in the decoder but not in the encoder. Thus, even for an observation $\mathbf{x}$ from domain $k$, our encoder $f$ may have nonzero outputs on $\mathcal{I}_{k'}$ (thus, $\mathbf{z}_{k'}$ is not necessarily 0) for $k \neq k'$. However, the sparsity regularizer $r(\cdot)$ encourages the encoder to parsimoniously use each dimension in the latent space so that there won't be much redundancy in the representation. We will show that an appropriately-designed sparsity regularizer can encourage the use of shared factor.

**Identifiability** We show that, when the decoding process in (1) is linear, an autoencoder trained in our approach can identify the true private and shared factors up to linear transformations. Although our proof is done for $L = 2$, the strategy should be extendable to the case with $L > 2$.

**Proposition 2.1.** *Suppose that the datasets in two domains*

$$X_k = [x_k^{(1)}; x_k^{(2)}; ...; x_k^{(n_1)}] \in \mathbb{R}^{n_k \times d}, \quad k = 1, 2 \tag{3}$$

*are generated from the latents $Z_k \in \mathbb{R}^{n_k \times m}$ as*

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} W := \begin{pmatrix} A & 0 & C_1 \\ 0 & B & C_2 \end{pmatrix} W \tag{4}$$

*where $W \in \mathbb{R}^{m \times d}$ maps $(A, C_1)$ and $(B, C_2)$ invertibly to $X_1$ and $X_2$ repsectively , while Ker$(C_1)$ = Ker$(C_2)$ and col$(A) \perp$ col$(C_1)$ as well as col$(B) \perp$ col$(C_2)$. Consider training another representation $X = \hat{Z}\hat{W}$ in the same form by minimizing the reconstruction loss as well as the sparsity loss*

$$r(Z) = \|\hat{A}\|_{\text{nzc}} + \|\hat{B}\|_{\text{nzc}} + \lambda\|\hat{C}\|_{\text{nzc}}, \quad 0 < \lambda < 2, \tag{5}$$

*where $\|\cdot\|_{\text{nzc}}$ is the number of nonzero columns in the matrix. Then the trained factors $(\hat{A}, \hat{B}, \hat{C}_1, \hat{C}_2)$ identifies $(A, B, C_1, C_2)$ upto mixing within each component.*

What does this proposition mean in terms of (2)? We note that the relation $x = W^\top z$ describes the nonparametric encoder $f : x \in \mathbb{R}^d \to z \in \mathbb{R}^m$ and the linear decoder $g : z \to W^T z$. Thus, in the stacked representation $Z \in \mathbb{R}^{(n_1+n_2)\times m}$ that appears in the relation (4), each column represents the latent coordinate dimension. If $A \in \mathbb{R}^{n_1\times m_a}$, $B \in \mathbb{R}^{n_2\times m_b}$, $C_k \in \mathbb{R}^{n_k\times m_c}$ so that $m_a + m_b + m_c = m$, then $m_a$ corresponds to the private factor dimension for $X_1$, $m_b$ corresponds to the private factor dimension for $X_2$, and $m_c$ corresponds to the shared factor dimension($m_{shared}$ in the previous section). Thus, the column sparseness regularization (5) encourages the latent space to be sparse in each factor dimension. We note that, in (4), the mask used in (2) is included the form of $Z_k$ itself. Also, the factor orthogonality is analogous to independency. Therefore, this result suggests a learning strategy that encourages independencies as well as sparseness in the encoded latent space while using a masked decoder that respects the private/shared structure.

**Implementation by VAE** We would like to train an autoencoder based on our finding in Proposition 2.1. Unfortunately, however, $\|\cdot\|_{nzc}$ is a variation of the $\ell_0$ norm and its direct optimization is NP-hard (Feng et al., 2018) in general. In our experiment, we propose to kill two birds with one stone by combining (2) with $\beta$-VAE (Higgins et al., 2016), which is known to encourage the sparsity as well as factor independencies (Rolinek et al., 2019; Zietlow et al., 2021). We put the details of VAE implementation in Appendix B.

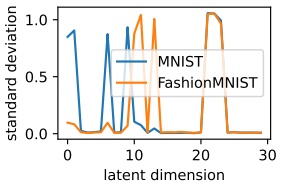
**Related Work** The idea of separating the latent space into private and shared ones has a long history. Canonical correlation analysis (Hotelling, 1936) captures a linear subspace where two random variables are mostly correlated, and the captured subspace can be seen as the space of shared factors. (Salzmann et al., 2010) extended this idea by simultaneously finding the private spaces. In the literature on domain adaptation and generalization, a lot of studies have used the shared-private representation, based on various approaches such as linear orthogonality (Bousmalis et al., 2016), adversarial frameworks (Bousmalis et al., 2016; Liu et al., 2017; Cao et al., 2018; Peng et al., 2019), and meta learning (Bui et al., 2021). Zhang et al. (2020) generalized the VAE framework to cover a broad family of graphical models where the variable relationships are described by the mask. As the closest work, Chattopadhyay et al. (2020) adopted the binary mask for the shared-private decomposition. Unlike us, they incorporate the space separation by learning the masks. In this study, we empirically and theoretically explore how the sparseness regularization helps the identification of the true private/shared structure in an unsupervised manner.

Some recent studies have utilized the sparseness in deep generative models as a form of domain knowledge in datasets such as videos (Klindt et al., 2020) and texts (Moran et al., 2021). Khemakhem et al. (2020) show that nonlinear VAE is identifiable when the prior distribution cleverly leverages auxiliary information such as class labels. Our case is related to this study if we see the domain label as auxiliary information. Unfortunately, we cannot directly compare their analysis to ours because they assume $m$ to be smaller than the number of domains.

## 3. Experiments

As a proof of concept, we use two datasets; `MNIST` as domain 1 and Fashion MNIST (`FMNIST`) as domain 2. To ensure that they share some common factors, we expand the image size from $28 \times 28$ to $64 \times 64$ and apply random 2D translation to both domains. As the private factors, we apply random rotation to `MNIST` and color randomization to `FMNIST`. We train our model for 100 epochs by Adam of batch size 32 and learning rate $2e-4$. We set $\beta = 3$ and the total number of factors to 30 so that we have 10 for each private and shared factor. We employ a three-layer convolution network as the encoder and spatial broadcasting (Watters et al., 2019) as the decoder.

**Latent Traversal** First, we study the active/inactive coordinates of the learned features for the observations in each domain. The right figure shows the domain-wise standard deviation of $\mathbf{z}$ for the test data. It clearly shows 1) roughly half of the features do not vary on both datasets and 2) the encoder learns the active/inactive patterns enforced by the mask used for the decoder. Figure 3 depicts the latent traversals over the factors that are relevant to the four augmentations we used to generate the dataset.[1] As expected, the 2D translation is captured as the shared factors while rotation and colorization are captured as private factors.




Next, we compare how the masking affects the learned representation. We train the vanilla $\beta$-VAE with the same setting and obtain the latent factors that are the most sensitive in reconstruction. To select the features based on sensitivity, we approximate the Jacobian of the decoder by the finite difference method and select the top-two factors in terms of the $\ell_2$ norm. Figure 4 compares the two-dimensional latent traversals of an encoded `MNIST` input for our method and $\beta$-VAE. We see that the variations in the proposed latent subspace are closed within `MNIST`, while $\beta$-VAE inadvertently captures the across-domain axis that connects digits

[1] The full results are shown in Appendix C

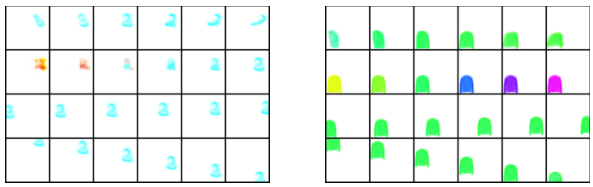


*Figure 3.* Latent traversals corresponding to rotation, color, x-translation, and y-translation (from top to bottom) for an MNIST input (left) and a FashionMNIST input (right).

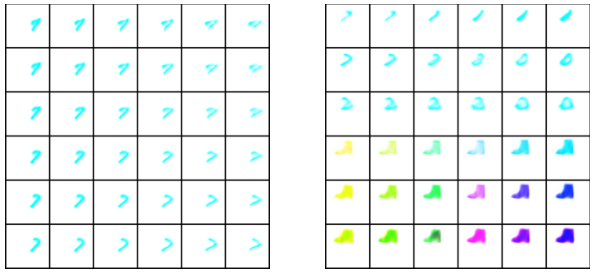


*Figure 4.* Two-dimensional latent traversals of the top-two reconstruction-sensitive latent factors. Left: proposed. Right: vanilla $\beta$-VAE.

and clothing shapes, exhibiting the same faulty behavior of PCA explained in the introduction.

**Counterfactual Generation** If the proposed method succeeds in identifying the private and shared factors, we shall be able to use the private factor of domain 1 to alter a dataset in domain 2 to produce counterfactual data without corrupting the contextual meaning of the domain1-private factor. To validate this, we study how the image transforms when we change the mask in the reconstruction process. For example, let $\mathbf{u}_{\mathrm{ms}} = [\mathbf{1}, \mathbf{0}, \mathbf{1}]$ be the mask for `MNIST`. If we apply $\mathbf{u}_{\mathrm{ms}}$ to $\mathbf{z}$ of an `FMNIST` input, the appearance of the reconstruction should look like `FMNIST` while keeping the position, since *position* is used as the shared factors in the generation process. The third row of Figure 5 shows the reconstruction results with $\mathbf{u}_{\mathrm{ms}}$. As expected, while the private information (`FMNIST` shapes and color) is overwritten, the shared information (position) is preserved. Similarly, the fourth and fifth rows show the reconstructions when we use the mask that deletes the shared factors and preserves the private factors. The positional information is lost in these figures, again confirming our hypothesis.

**Domain Adaptation** Finally, we check how the obtained representation is useful in a domain adaptation situation. Here we consider an unsupervised case where we have many data in the source domain but a few in the target domain. We first train the VAE, and use the trained encoder as the

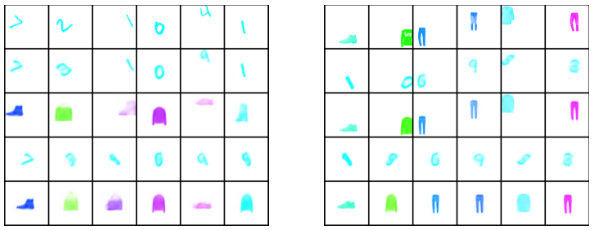


*Figure 5.* Counterfactual generation for MNIST (left) and FMNIST (right) inputs. From top to bottom: raw inputs, reconstructions with the (MNIST-private+shared) mask, (FMNIST-private+shared) mask, MNIST-private-only mask , and FMNIST-private-only mask.

fixed feature extractor. We then evaluate the classification accuracy by a linear probe setting, i.e., add a linear layer and minimize the softmax cross-entropy loss. We use `MNIST` as the source and `FMNIST` as the target, where we apply random augmentations in the form of horizontal translation to `MNIST` and vertical translation to `FMNIST`. We apply random rotation to both datasets. We conducted a `FMNIST` classification task on two settings: IID and OOD. In IID, we only apply the random rotation and random horizontal translation to the test set of `FMNIST`. In OOD, We apply all random transformations ( rotation, horizontal translation, vertical translation) to the test set of `FMNIST`. Figure 6 compares the accuracy of the vanilla $\beta$-VAE and the proposed method for different proportions of `FMNIST` used in the training of the VAE. The proposed method outperforms the original VAE in both IID and OOD settings, especially when the number of target domain data is small.

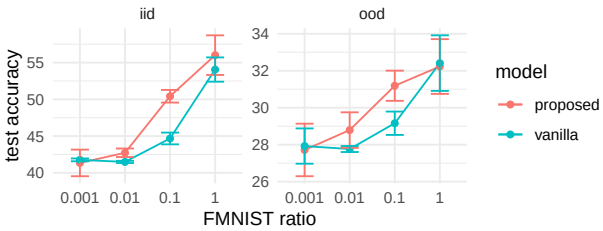



*Figure 6.* Linear probe performance. The test accuracy on `FMNIST` is plotted against the data proportion #`FMNIST`/ #`MNIST` in the training set.

**Conclusion** In this paper, we explored the way to enforce the private/shared decomposition in the latent space by using the masked decoder and sparsity-regularized encoder. The empirical and theoretical results in this paper suggest that there is much more room left to research on the way of enforcing the desired structure in unsupervised learning. Future works include the extension of the theory to a non-linear decoder and a similar investigation of the other forms of structures that are useful in domain adaptation.

## References


Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., and Erhan, D. Domain separation networks. *Advances in neural information processing systems*, 29, 2016.

Bui, M.-H., Tran, T., Tran, A., and Phung, D. Exploiting domain-specific features to enhance domain generalization. *Advances in Neural Information Processing Systems*, 34, 2021.

Cao, J., Katzir, O., Jiang, P., Lischinski, D., Cohen-Or, D., Tu, C., and Li, Y. Dida: Disentangled synthesis for domain adaptation. *arXiv preprint arXiv:1805.08019*, 2018.

Chattopadhyay, P., Balaji, Y., and Hoffman, J. Learning to balance specificity and invariance for in and out of domain generalization. In *European Conference on Computer Vision*, pp. 301–318. Springer, 2020.

Feng, M., Mitchell, J. E., Pang, J.-S., Shen, X., and Wächter, A. Complementarity formulations of l0-norm optimization problems. *Pacific Journal of Optimization*, 14(2): 273–305, 2018.

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

Hotelling, H. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936. ISSN 00063444.

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

Klindt, D., Schott, L., Sharma, Y., Ustyuzhaninov, I., Brendel, W., Bethge, M., and Paiton, D. Towards nonlinear disentanglement in natural data with temporal sparse coding. *arXiv preprint arXiv:2007.10930*, 2020.

Liu, P., Qiu, X., and Huang, X. Adversarial multi-task learning for text classification. *arXiv preprint arXiv:1704.05742*, 2017.

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

Moran, G. E., Sridhar, D., Wang, Y., and Blei, D. M. Identifiable deep generative models via sparse decoding. *arXiv preprint arXiv:2110.10804*, 2021.

Peng, X., Huang, Z., Sun, X., and Saenko, K. Domain agnostic learning with disentangled representations. In *International Conference on Machine Learning*, pp. 5102–5112. PMLR, 2019.

Rolinek, M., Zietlow, D., and Martius, G. Variational autoencoders pursue pca directions (by accident). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12406–12415, 2019.

Salzmann, M., Ek, C. H., Urtasun, R., and Darrell, T. Factorized orthogonal latent spaces. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 701–708. JMLR Workshop and Conference Proceedings, 2010.

Watters, N., Matthey, L., Burgess, C. P., and Lerchner, A. Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes. *arXiv preprint arXiv:1901.07017*, 2019.

Zhang, R., Koyama, M., and Ishiguro, K. Learning structured latent factors from dependent data: A generative model framework from information-theoretic perspective. In *International Conference on Machine Learning*, pp. 11141–11152. PMLR, 2020.

Zietlow, D., Rolinek, M., and Martius, G. Demystifying inductive biases for $\beta$-vae based architectures. *arXiv preprint arXiv:2102.06822*, 2021.

# A. Proof and the formal statement of 2.1

*Proof of Proposition 2.1.* Let $U^+$ denote the pseudo inverse of matrix $U$. Put $W\hat{W}^+ = P$ so that

$$\begin{pmatrix} A & 0 & C \\ 0 & B & C \end{pmatrix} \begin{pmatrix} P_{aa} & P_{ab} & P_{ac} \\ P_{ba} & P_{bb} & P_{bc} \\ P_{ca} & P_{cb} & P_{cc} \end{pmatrix} = \begin{pmatrix} AP_{aa} + C_1P_{ca} & AP_{ab} + C_1P_{cb} & AP_{ac} + C_1P_{cc} \\ BP_{ba} + C_2P_{ca} & BP_{bb} + C_2P_{cb} & BP_{cb} + C_2P_{cc} \end{pmatrix} = \begin{pmatrix} \hat{A} & 0 & \hat{C} \\ 0 & \hat{B} & \hat{C} \end{pmatrix} \tag{6}$$

We will show that all but diagonal blocks of $P$ are zero. Since $W\hat{W}^+$ is invertible on its restriction to both $(A, C)$ and $(B, C)$,

$$\begin{aligned} \text{rank}(A) + \text{rank}(C_1) &= \text{rank}(\hat{A}) + \text{rank}(\hat{C}_1) \\ \text{rank}(B) + \text{rank}(C_2) &= \text{rank}(\hat{B}) + \text{rank}(\hat{C}_2) \end{aligned} \tag{7}$$

Also, the rank agrees with $\|\cdot\|_0$ in its sparsest form for $A, B$, and this applies to $C$ as well because $\lambda > 0$. Thus, by the minimality assumption,

$$\text{rank}(A) + \text{rank}(B) + \lambda\text{rank}(C_1) \geq \text{rank}(\hat{A}) + \text{rank}(\hat{B}) + \lambda\text{rank}(\hat{C}_1). \tag{8}$$

Put

$$\Delta a = \text{rank}(A) - \text{rank}(\hat{A}) \ \ , \Delta b = \text{rank}(B) - \text{rank}(\hat{B}), \ \ \Delta c_k = \text{rank}(C_k) - \text{rank}(\hat{C}_k)$$

Since $\text{rank}(C_2) = \text{rank}(C_1)$, we use $\Delta c$ to denote $\Delta c_k$. then note that

$$\begin{aligned} &(\text{rank}(A) + \lambda\text{rank}(C_1)) - (\text{rank}(\hat{A}) + \lambda\text{rank}(\hat{C}_1)) && (9) \\ &= (\text{rank}(A) + \text{rank}(C_1)) - (\text{rank}(\hat{A}) + \text{rank}(\hat{C}_1)) + (\lambda - 1)\Delta c && (10) \\ &= (\text{rank}(A) + \text{rank}(C_1)) - (\text{rank}(A) + \text{rank}(C_1)) + (\lambda - 1)\Delta c && (11) \\ &= (\lambda - 1)\Delta c && (12) \end{aligned}$$

Using this relation we obtain

$$\begin{aligned} \text{rank}(A) + \text{rank}(B) + \lambda\text{rank}(C) &\geq \text{rank}(\hat{A}) + \text{rank}(\hat{B}) + \lambda\text{rank}(\hat{C}) && (13) \\ (\lambda - 1)\Delta c &= \text{rank}(\hat{B}) - \text{rank}(B) && (14) \\ &\geq -\Delta b \geq \Delta c && (15) \\ \lambda\Delta c &\geq 2\Delta_c && (16) \end{aligned}$$

If $\lambda < 2$, this holds only if $\Delta c = 0$. By (7), this forces $\Delta a = 0$, $\Delta b = 0$. Recall that we have

$$AP_{ac}^T + C_1P_{cc}^T = \hat{C}_1, \ \ BP_{bc}^T + C_2P_{cc}^T = \hat{C}_2$$

Now, by assumption, $\text{Ker}(C_1) = \text{Ker}(C_2)$. Thus, for exactly $k_0$ number of nonzero $v$, we have

$$\begin{aligned} \hat{C}_1 v &= (AP_{ac}^T + C_1P_{cc}^T)v = 0 \\ \hat{C}_2 v &= (BP_{bc}^T + C_2P_{cc}^T)v = 0 \end{aligned} \tag{17}$$

However this implies that

$$AP_{ac}^T v = -C_1P_{cc}^T v \quad BP_{bc}^T v = -C_2P_{cc}^T v$$

But since $\text{col}(A) \perp \text{col}(C_1)$ as well as $\text{col}(B) \perp \text{col}(C_2)$, this would take place only if

$$C_1P_{cc}^T v = 0, \quad C_2P_{cc}^T v = 0$$

Combining this result with $\Delta c = 0$, we therefore have

$$\text{rank}(C_1) = \text{rank}(\hat{C}_1) = \text{rank}(C_1P_{cc}^T)$$

Because the presence of $AP_{ac}^T$ only increases the rank in the (17) by orthogonality, $\text{rank}(C_1 P_{cc}^T) = \text{rank}(C_1)$ implies $AP_{ac}^T = 0$. Most importantly, $\text{col}(C_1) = \text{col}(\hat{C}_1)$ as spaces. Likewise, $BP_{bc}^T = 0$ and $\text{col}(C_2) = \text{col}(\hat{C}_2)$.

Finally, recall that

$$\begin{aligned} \hat{A} &= AP_{aa}^T + C_1 P_{ca}^T \\ \hat{B} &= BP_{bb}^T + C_2 P_{cb}^T \end{aligned} \tag{18}$$

Baecause $\text{col}(C_1) = \text{col}(\hat{C}_1)$, the column space of $\hat{A}$ cannot intersect with $\text{col}(C_1)$ by the orthogonality relation $\text{col}(\hat{C}_1) \perp \text{col}(\hat{A})$. Thus $C_1 P_{ca}^T = 0$. Likewise, $C_2 P_{cb}^T = 0$ with symmetrical argument. Alltogether,

$$\begin{pmatrix} \hat{A} & 0 & \hat{C} \\ 0 & \hat{B} & \hat{C} \end{pmatrix} = \begin{pmatrix} AP_{aa} + C_1 P_{ca} & AP_{ab} + C_1 P_{cb} & AP_{ac} + C_1 P_{cc} \\ BP_{ba} + C_2 P_{ca} & BP_{bb} + C_2 P_{cb} & BP_{cb} + C_2 P_{cc} \end{pmatrix} \tag{19}$$

$$= \begin{pmatrix} AP_{aa} & 0 & C_1 P_{cc} \\ 0 & BP_{bb} & C_2 P_{cc} \end{pmatrix} \tag{20}$$

$$= \begin{pmatrix} A & 0 & C \\ 0 & B & C \end{pmatrix} \begin{pmatrix} P_{aa} & 0 & 0 \\ 0 & P_{bb} & 0 \\ 0 & 0 & P_{cc} \end{pmatrix} \tag{21}$$

And the claim follows.

□

## B. Detail of VAE implementation

VAE employs the variational distribution $q(\mathbf{z} \mid \mathbf{x}) = N(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}^2))$ as a posterior, where the mean and the variance are obtained by the encoders $f_\mu, f_{\sigma^2}$ as $\boldsymbol{\mu} = f_\mu(\mathbf{x})$ and $\boldsymbol{\sigma}^2 = f_{\sigma^2}(\mathbf{x})$. By using the reparametrization trick $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma}\boldsymbol{\epsilon}$ with a random Gaussian $\boldsymbol{\epsilon} \sim N(0, I)$, the $\beta$ VAE objective with the Gaussian likelihood is given by Eq. 2 where the regularization term corresponds to the KL divergence: $r(\mathbf{z}) = \text{KL}(p(\mathbf{z}) \| q(\mathbf{z}|\mathbf{x})) = \frac{1}{2}\sum_{i=1}^m (\mu_i^2 + \sigma_i^2 - \log \sigma_i^2 - 1)$. The KL term measures the distance between the standard Gaussian prior $p(\mathbf{z}) = N(0, I)$ and the variational distribution. Essentially, we simply use this KL term as the choice of $r$ in (2). We do not use the masking structure for the encoder itself because the mask in the decoding process would make sure that the reconstruction loss of the domain $k$ is not affected by $\mathbf{z}'_k$ with $k \neq k'$. For example, suppose that $\mathbf{z}_2$ is nonzero for the generation of domain 1. In this case, $\mathbf{z}_2$ does not contribute to the reconstruction loss in (2), so the derivative of the reconstruction with respect to $\mathbf{z}_2$ is 0 and $q(\mathbf{z}_2|\mathbf{x}) = p(\mathbf{z}_2)$. Thus, optimizing the mean and variance parameter with respect domain 1 alone would automatically prefer $\boldsymbol{\mu}_2 = 0$ and $\boldsymbol{\sigma}_2^2 = 1$ as the optimal solution.

## C. Full results of latent traversals

Figure 7 shows the latent traversals of the proposed method for all the latent factors where the standard deviation of $\mathbf{z}$ is larger than $0.5$. Similarly, Figure 8 shows the latent traversals of $\beta$-VAE.

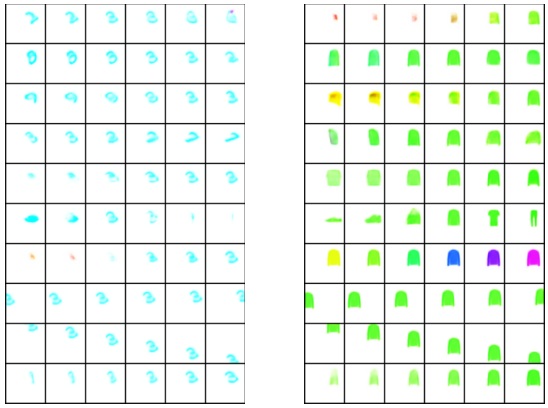


*Figure 7.* Latent traversals of the proposed method for an MNIST input (left) and a FashionMNIST input (right).

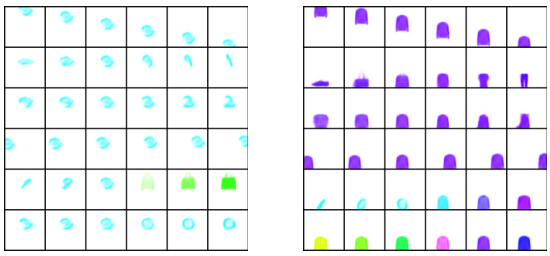


*Figure 8.* Latent traversals of $\beta$-VAE for an MNIST input (left) and a FashionMNIST input (right).