Multiple Invertible and Partial-Equivariant Function for Latent Vector Transformation to Enhance Disentanglement in VAEs

Hee-Jun Jung, Jaehyoung Jeong, Kangil Kim, *member, IEEE*

Abstract—Disentanglement learning is a core issue for understanding and re-using trained information in Variational AutoEncoder (VAE), and effective inductive bias has been reported as a key factor. However, the actual implementation of such bias is still vague. In this paper, we propose a novel method, called *Multiple Invertible and partial-equivariant transformation* (MIPE-transformation), to inject inductive bias by 1) guaranteeing the invertibility of latent-to-latent vector transformation while preserving a certain portion of equivariance of input-to-latent vector transformation, called *Invertible and partial-equivariant transformation* (IPE-transformation), 2) extending the form of prior and posterior in VAE frameworks to an unrestricted form through a learnable conversion to an approximated exponential family, called *Exponential Family conversion* (EF-conversion), and 3) integrating multiple units of IPE-transformation and EF-conversion, and their training. In experiments on 3D Cars, 3D Shapes, and dSprites datasets, MIPE-transformation improves the disentanglement performance of state-of-the-art VAEs.

Index Terms—Variational Auto-Encoder, Disentanglement Learning, Equivariant Function, Invertible Function, Partial equivariance.

1 Introduction

Disentanglement learning to learn more interpretable representations is broadly useful in artificial intelligence fields such as classification [1], zero-shot learning [2], and domain adaptation [3], [4]. The disentangled representation is defined as a change in a single dimension, which corresponds to unique semantic information. Several works have been conducted based on this framework.

A major model for enhancing the disentanglement learning is Variational AutoEncoder (VAE) [5]. Based on VAE, unsupervised disentangled representation learning has been elaborated [6]–[10] through the factorizable variations and control of uncorrelatedness of each dimension of representations. Moreover, VAE models that handle the shape of prior as a Gaussian mixture [11] or von Mises-Fisher [12] were also developed, but the disentanglement is still incomplete. As a critical point, there is a report that unsupervised disentanglement learning is impossible without inductive bias [13].

Recently, such inductive bias has been introduced in various perspectives on transformation of latent vector space. Intel-VAE [14] proposed the benefit of *invertible* transformation of the space to another latent space to provide better data representation, which includes hierarchical representations. Group theory based bias also shows significant improvement on disentanglement [15], [16], whose definition follows [17], which is based on the group theory. The works show that *equivariant* transformation between input and latent vector space has a key role of disentanglement.

H.-J. jung, J. Jeong, and K. Kim are with the AI Graduate School, Gwangju Institute of Science and Technology (GIST), Gwangju 61005, Republic of Korea (e-mail: {jungheejun93, jaehyoung98}@gm.gist.ac.kr, kikim01@gist.ac.kr). (Corresponding author: Kangil Kim.)

Inspired by the above works, we propose a *Multiple Invertible and partial-equivariant transformation* (MIPE-transformation) method, which is simply insertable to VAEs. The *partial-equivariance* is defined by [18] and we follow its definition: $f(g \cdot x) = g \cdot f(x)\ \forall x \in \mathcal{X}, \forall g \in G', G' \subset G$, where G' is a subset of group G (f is partially equivariant to group G). First, we assume that an encoder is *partial-equivariant* and we call it an *encoder equivariance condition*. The method adopts the matrix exponential to hold the invertible property of latent-to-latent (L2L) vector transformation. Then, we constrain the L2L transformation to a symmetric matrix exponential to be partial-equivariant to a subgroup between latent and transformed latent space. Because it extends the encoder to be partial-equivariant to a subgroup between input and transformed latent space. The IPE-transformation generates an uncertain form of latent vector distributions, so we provide a training procedure to force them to be close to an exponential family, called *exponential family conversion* (EF-conversion). This conversion enables the uncertain distribution to work in the typical training framework of VAEs. Then, we mathematically show that the multiple uses of IPE-transformation work as β parameters [6] controlled for enhancing disentanglement learning. Also, we propose the *implicit semantic mask* to induce a semantic mask in the latent vector space, different to [19]. In experiments with quantitative and qualitative analysis, MIPE-transformation shows significant improvement in disentangled representation learning in 3D Cars, 3D Shapes, and dSprites tasks. Our main contributions are summarized as follows.

1) We propose to use a symmetric matrix exponential as a latent-to-latent vector transformation function for inducing inductive bias based on invertible and equivariant properties with mathematical analysis.

2) We provide a training procedure and losses for VAEs to learn unknown latent vector distribution as an approximated exponential family.
3) We propose the novel MIPE-transformation architecture to integrate multiple IPE-transformation and EF-conversion, which is widely applicable to state-of-the-art VAEs.
4) We empirically analyze the properties of MIPE-transformation and validate its effectiveness in disentanglement learning on benchmarks.

2 Related Work

Recently, various works have focused on the unsupervised disentanglement learning. Previous works are based on [20] definition. One of the branches is InfoGAN [21] based works such as IB-GAN [22] implement extra regularizer to improve informativeness [23]. The other branch is based on the VAE. β-VAE [6] penalizes Kullback-Leibler divergence (KL divergence) with weighted hyper-parameters. Factor VAE [8] and β-TCVAE [7] are trained with total correlation (TC) to make independent dimensions on a latent vector with discriminator and divided components of KL divergence term. Differently, we consider the recent disentanglement definition based on group theory [17].

Following the definitions of disentangled representation learning by group theory, several works have emphasized equivariant and improved disentangled representation learning. Commutative Lie Group VAE (CLG-VAE) [15] proposed direct mapping of the latent vector into Lie algebra to obtain group structure (inductive bias) with constraints: commutative and hessian loss. Furthermore, Groupified VAE [16] utilizes Spatial Broadcast Decoder [24] to implement an equivariant function to the cyclic group with guaranteeing commutativity and invertibility of group actions. Topographic VAE [25] combines Student's-t distributions and variational inference. It enforces rotated latent vectors to be equivariant. On the other hand, we apply unrestricted prior and posterior for disentanglement learning.

There are several inductive biases to learning unsupervised disentanglement, such as group theory based and sequential order. In this section, we briefly discuss sequential order inductive bias even though its method is considered in different domains such as text and video frames. To individualize the static (time-invariant) and dynamic (time-variant), [26], [27] proposed the latent variables one (f) is only dependent on the given times series datasets $x_{1:T}$, and the other ($\mathbf{z}_{1:T}$) is dependent on the $x_{1:T}$ and f. Moreover [27] propose the novel ELBO with maximizing mutual information between the input and the latent vectors. These works empirically show that sequential order which includes separated latent vectors improves unsupervised disentanglement learning with diverse qualitative analysis. Differently in group theory based approaches, the proposed methods consider equivariant function between input and latent vector space.

Other VAE approaches implement other prior from Gaussian distribution to transformed Gaussian distribution, Gaussian mixture distribution [28] or von Mises-Fisher distribution [12]. InteL-VAE [14] shows that transformed Gaussian distribution by the invertible function trains hierarchical representation with manual function. We show more clear relation of invertibility to disentanglement and improve VAEs to use its unrestricted form of prior.

Invertible and equivariant Deep Neural Networks have been investigated with normalizing flows. As proven by [29], utilized matrix exponential on Neural networks is invertible, but it only provides mathematical foundations of the tra nsformation. Matrix exponential is utilized to implement an invertible and equivariant function to improve the generative flow compare to linear function [30]. To specify the exponential familyt, other works contribute uncertainty of exponential family distribution with Bayesian update [31], [32]. In addition, [33] hierarchically controls the natural parameter across the layers and determines the exponential family distribution with the moment of sufficient statistic. In our work, we show how to use it for disentanglement learning.

3 Preliminaries

3.1 Group Theory

Binary operation: Binary operation on a set S is a function that $* : S \times S \rightarrow S$, where $\times$ is a cartesian product.

Group: A group is a set G together with binary operation $*$, that combines any two elements g_a and g_b in G, such that the following properties:

- closure: $g_a, g_b \in G \Rightarrow g_a * g_b \in G$.
- Associativity: $\forall g_a, g_b, g_c \in G$, *s.t.* $(g_a * g_b) * g_c = g_a * (g_b * g_c)$.
- Identity element: There exists an element $e \in G$, *s.t.* $\forall g \in G, e * g = g * e = g$.
- Inverse element: $\forall g \in G, \exists g^{-1} \in G$: $g * g^{-1} = g^{-1} * g = e$.

Group action: Let $(G, *)$ be a group and set X, binary operation $\cdot : G \times X \rightarrow X$, such that following properties:

- Identity: $e \cdot x = x$, where $e \in G, x \in X$.
- Compatibility: $\forall g_a, g_b \in G, x \in X, (g_a * g_b) \cdot x = g_a \cdot (g_b \cdot x)$.

Equivariant map: Let G be a group and X_1, X_2 be two sets with corresponding group action of G in each sets: $T_g^{X_1}, T_g^{X_2}$, where $g \in G$. Then a function $f : X_1 \rightarrow X_2$ is equivariant if $f(T_g^{X_1} \cdot X_1) = T_g^{X_2} \cdot f(X_1)$.

Partial Equivariance: [18]: Let subset of G be $\Upsilon \subset G$, then f is a partially equivariant map to G:

$$f(T_\upsilon^{X_1} \cdot X_1) = T_\upsilon^{X_2} \cdot f(X_1), \text{ where } \forall \upsilon \in \Upsilon. \tag{1}$$

Homomorphsim: Let $(G, \cdot), (H, \circ)$ be two groups. If mapping function $h : G \rightarrow H$, *s.t.* $h(g_i \cdot g_j) = f(g_i) \circ f(g_j)$, then f is called homomorphism.

3.2 Exponential Family

Power density function of the exponential family (PDF) generalized formulation:

$$\begin{aligned} f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}) &= h(\boldsymbol{x})\exp(\boldsymbol{\theta}^\intercal T(\boldsymbol{x}) - A(\boldsymbol{\theta})) \\ &= \exp(\boldsymbol{\theta}^\intercal T(\boldsymbol{x}) - A(\boldsymbol{\theta}) + B(\boldsymbol{x})), \end{aligned} \tag{2}$$

where *sufficient statistics* $T(\cdot)$, *log-normalizer* $A(\cdot)$, and *carrier or base measure* $B(\cdot)$ are known functions, samples $\boldsymbol{x}$ from distribution, and *natural parameter* $\boldsymbol{\theta}$.

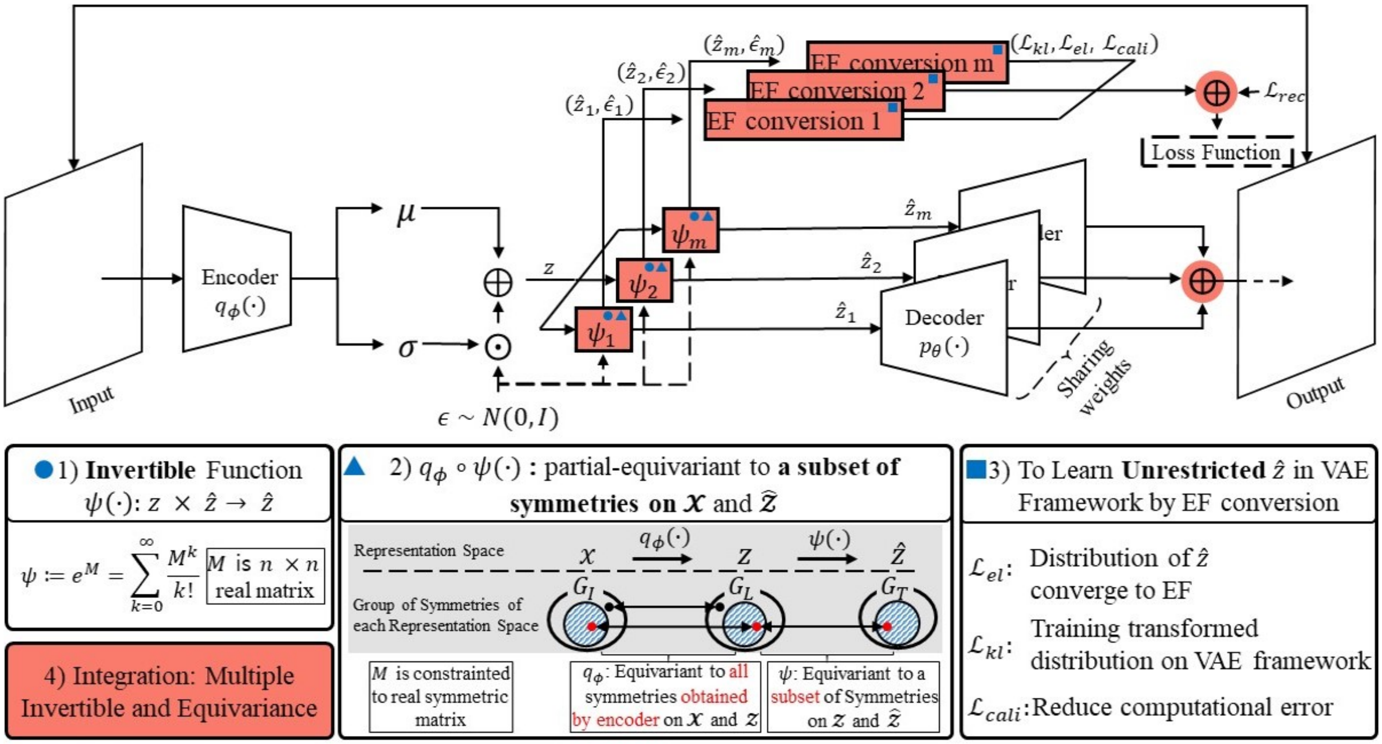


Fig. 1: The overall architecture of our proposed *MIPET*-VAE. The invertible and partial-equivariant function $\psi(\cdot)$ for L2L transformation consists of a symmetric matrix exponential to be 1) invertible and 2) partial-equivariant. Then 3) EF conversion module converges the distribution of unrestricted $\hat{z}$ to be EF with $\mathcal{L}_{el}$ loss. Also, it applies KL divergence loss ($\mathcal{L}_{kl}$) between the transformed posterior and prior, which are expressed by the power density function of EF. In the last, EF conversion reduces the computational error ($\mathcal{L}_{cali}$) between approximated and true KL divergence. 4) The reddish color represents the integration parts. The blue figures represent each property. The details of the gray box are in Figure 2.

4 METHOD

The overview of a VAE equipped with MIPE-transformation is shown in Figure 1. The MIPE-transformation has three main components: 1) *IPE-transformation Unit* to transform latent vectors with invertible and partial-equivariant properties, 2) *EF-conversion Unit* to extend VAEs to learn the exponential family distribution of latent vectors, and 3) integrated training and generation process for multiple uses of IPE-transformation and EF-conversion.

4.1 Invertible and Partial-Equivariant Function for L2L Transformation

4.1.1 Invertible Property by Using Matrix Exponential

To guarantee the invertible property of IPE-transformation, we use a function $\psi(\cdot) = \mathbf{e}^{\boldsymbol{M}} * \cdot$ for the transformation, where $\boldsymbol{M}$ is in $n \times n$ real number matrix set $M_n(\mathbb{R})$ [29]. The operator $*$ is matrix multiplication, and $\mathbf{e}^{\boldsymbol{M}} = \sum_k^\infty \frac{\boldsymbol{M}^k}{k!}$. Our motivation is to use the benefits of injecting explicit inductive bias for disentanglement [13], [14]. InteL-VAE effectively extracts hierarchical representation, which includes low-level features (affect to a specific factor) and high-level features (affect to complex factors) with an invertible transformation function [14].

4.1.2 Why Should L2L Transformation Be Equivariant?

Let's consider equivariant function between the input and transformed latent vector space, directly used for a decoder in the VAE frameworks. All L2L transformations do not extend the encoder equivariance condition to the relation between input and transformed latent space. This problem is more precisely shown in Figure 2, which illustrates partial equivariance condition over the input space $\mathcal{X}$, latent vector space $\mathcal{Z}$, and its transformed latent vector space $\hat{\mathcal{Z}}$ with a corresponding group of symmetries G_I, G_L, and G_T, respectively. In the VAEs literature, it has not been reported to restrict L2L transformation to guarantee equivariant function between two spaces, so we propose a solution to guarantee at least a part of symmetries to be equivariant.

4.1.3 Equivariance Property with Symmetric Matrix Exponential

To enhance the equivariance of L2L transformation, we set $\boldsymbol{M}$ of $\psi(\cdot)$ to a symmetric matrix. We show that 1) a group with the constraint guarantees equivariance of $\psi(\cdot)$ over the specific group, 2) $\psi(\cdot)$ being equivariant over subset of symmetries between the input space and transformed latent vector space, and 3) the constraint increases the probability of $\psi(\cdot)$ to be in the group (equal to be equivariant over the subset of symmetries).

We particularly call the transformations as *symmetries* [34] to distinguish them from IPE- and I2L-transformations. For the generality of our method, we consider an arbitrary VAE model that has no restriction on creating intersections to any set as Figure 2.

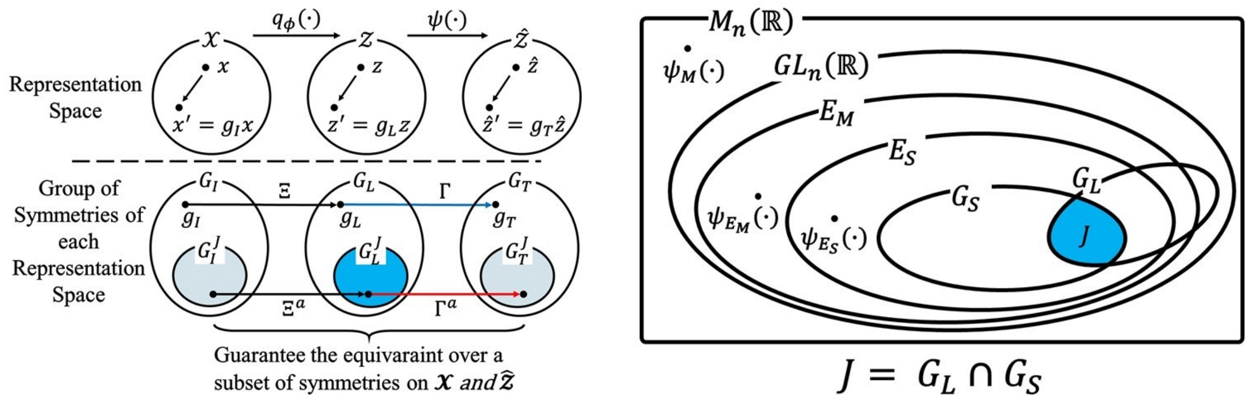


Fig. 2: G_I, and G_L are obtained through encoder q_ϕ (encoder equivariance condition). The left side figure shows the relation between each space and symmetries. If $\psi(\cdot)$ is equivariant function over all G_L, and G_T, then there exist Γ, where $\Gamma : G_L \to G_T$, and $\Xi \circ \Gamma : G_I \to G_T$. However, unrestricted $\psi(\cdot)$ has no guarantee to be partial- or full-equivariant. The red arrows represent our method: L2L transformation guarantees $\Gamma^J : G_L^J \to G_T^J$, and $\Xi^J \circ \Gamma^J : G_I^J \to G_T^J$, given the encoder equivariance condition $\Xi : G_I \to G_L$.

TABLE 1: Terms and Notations

$\boldsymbol{z}$	Latent vector from encoder
$\psi(\cdot)$	Invertible function
$\hat{\boldsymbol{z}}_m$	Transformed latent vector by $\psi_m(\cdot)$
$\hat{\boldsymbol{\epsilon}}_m$	Transformed prior samples by $\psi_m(\cdot)$
$\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m}$	Natural Parameter of posterior
$\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}_m}$	Natural Parameter of prior
T	Sufficient Statistics
A	Log-Normalizer
$\boldsymbol{\nu}$	Evidence
$D_{\mathrm{KL}}(\cdot\|\|\cdot)$	Kullback-Leibler divergence
$f_{\boldsymbol{x}}(\cdot)$	Power Density Function
$M_n(\mathbb{R})$	A set of $n \times n$ real matrix
$GL_n(\mathbb{R})$	General Linear Group
$Sym_n(\mathbb{R})$	A set of $n \times n$ symmetric real matrix
E_M	$\{\mathbf{e}^M \| M \in M_n(\mathbb{R})\}$
E_S	$\{\mathbf{e}^S \| S \in Sym_n(\mathbb{R})\}$
G_S	$G_S : (\mathbf{e}^S, *)$
G_I	Group of input space for symmetries
G_L	Group of latent space for symmetries
J	$G_S \cap G_L$
$\psi_M(\cdot)$	$\psi_M(\cdot) \in M_n(\mathbb{R})$
$\psi_{E_M}(\cdot)$	$\psi_{E_M}(\cdot) \in E_M$
$\psi_{E_S}(\cdot)$	$\psi_{E_S}(\cdot) \in E_S$
$\mathbf{0}$	zero vector
$\mathbf{0}_{n,n}$	n by n zero matrix
$\mathcal{X}$	Input space
$\mathcal{Z}$	Latent vector space
$\hat{\mathcal{Z}}$	Transformed latent vector space
Ξ	$G_I \times G_L \to G_L$
Γ	$G_L \times G_T \to G_T$
Ξ^J	$G_I^J \times G_L^J \to G_L^J$
Γ^J	$G_L^J \times G_T^J \to G_T^J$

In the next, we show that matrix exponential with symmetric matrix partially preserves encoder equivariance condition better than other matrices.

Proposition 4.1. *Any $\psi(\cdot) \in G_S$, notated as $\psi_{G_S}(\cdot)$, is equivariant to group G_S.*

Proof. The group G_S is closed to matrix multiplication, and its element is always a symmetric matrix by definition. Then, any two elements in G_S are commutative because if matrix multiplication of two symmetric matrices is symmetric then both are commutative. As a result, $\psi_{G_S}(\cdot)$ and group elements of G_S are commutative (G_S is an abelian group). Because of the commutativity, $\psi_{G_S}(g_s \circ \boldsymbol{z}) = \mathbf{e}^{\boldsymbol{S}} g_s \boldsymbol{z} = g_s \mathbf{e}^{\boldsymbol{S}} \boldsymbol{z} = g_s \circ \psi_{G_S}(\boldsymbol{z})$ for $g_s \in G_S$ if the group action $\circ$ is set to matrix multiplication, where $\psi_{G_S} \in G_S$. This equation satisfies the general definition of an equivariant function that a function $f(\cdot)$ is equivariant if $f(g \circ \boldsymbol{z}) = g \circ f(\boldsymbol{z})$ for all g in a group G by matching f, g, and G to ψ_{G_S}, g_s, and G_S, respectively. ■

Proposition 4.2. *If q_ϕ is equivariant over defined on group of symmetries G_I^a and G_L^a, then $\psi_{G_S}(q_\phi(\cdot))$ is equivariant to symmetries in G_I corresponding to $G_S \cap G_L$ and G_T corresponding to $G_S \cap G_L$ by the equivariance of q_ϕ.*

Proof. The function $\psi_{G_S}(\cdot)$ is an equivariant function over group elements in $G_S \cap G_L$ by Proposition 4.1. Then, the composite function, $\psi_{G_S}(\cdot)$ and q_ϕ, is an equivariant function of G_I corresponding to $G_S \cap G_L$ and G_T corresponding to $G_S \cap G_L$. Let g_L^a be a group element in $G_S \cap G_L$, and g_I^a is a group element in G_I corresponding to $G_S \cap G_L$, and g_T^a is a group element where corresponding to $G_S \cap G_L$ on the latent vector space transformed from the original latent vector space. Then, group element g_T^a is equal to g_L^a:

$$\hat{\boldsymbol{z}}_1 = \psi_{G_S}(\boldsymbol{z}_1), \text{and} \tag{3}$$

$$\hat{\boldsymbol{z}}_2 = \psi_{G_S}(\boldsymbol{z}_2) = \psi_{G_S}(g_L^a \boldsymbol{z}_1) = g_L^a \psi_{G_S}(\boldsymbol{z}_1) \ (\because \text{Prop. 4.1}), \tag{4}$$

$$\begin{aligned} &\text{then } g_L^a \psi_{G_S}(\boldsymbol{z}_1) = g_T^a \psi_{G_S}(\boldsymbol{z}_1) \ (\because \hat{\boldsymbol{z}}_2 = g_T^a \hat{\boldsymbol{z}}_1) \\ &\Rightarrow (g_L^a - g_T^a)\psi_{G_S}(\boldsymbol{z}_1) = \mathbf{0}, \end{aligned} \tag{5}$$

where $\mathbf{0}$ is a zero vector. Eq. 5 is defined when $\forall \boldsymbol{z} \in \mathcal{Z}$ by the equivariance definition. In other words, Eq. 5 is satisfied only if the kernel (linear algebra) of $g_L^a - g_T^a$, notated as *ker*$(g_L^a - g_T^a)$, includes the basis of $\mathbb{R}^n$ vector space. If the standard basis of $\mathbb{R}^n$ vector space is in *ker*$(g_L^a - g_T^a)$, then $(g_L^a - g_T^a) = \mathbf{0}_{n,n}$, where $\mathbf{0}_{n,n}$ is an n by n zero matrix. Other bases of $\mathbb{R}^n$ vector space are expressed by the standard basis. Therefore $g_L^a - g_T^a = \mathbf{0}_{n,n}$.

Then, $\psi_{G_S}(g_L^a \boldsymbol{z}_1) = g_L^a \psi_{G_S}(\boldsymbol{z}_1) = g_T^a \psi_{G_S}(\boldsymbol{z}_1)$. The encoder is an equivariant function over input space $\mathcal{X}$ as $q_\phi(g_I^a \boldsymbol{x}_1) = g_L^a q_\phi(\boldsymbol{x}_1)$. Mixing two equivarience property, we

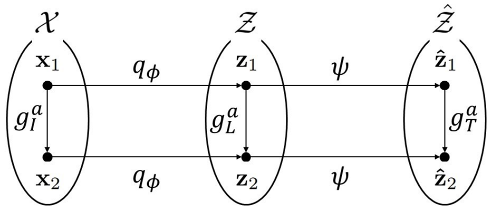


Fig. 3: Equivariant map: $\mathcal{X}$, $\mathcal{Z}$, and $\hat{\mathcal{Z}}$ are input space, latent vector space, and transformed latent vector space by L2L transfomration function $\psi(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^n$. respectively. $\boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{z} \in \mathcal{Z}$, and $\hat{\boldsymbol{z}} \in \hat{\mathcal{Z}}$.

can derive another equivariance relation $g_T^a \psi_{G_S}(q_\phi(\boldsymbol{x}_1)) = \psi_{G_S}(q_\phi(g_I^a \boldsymbol{x}_1))$ This result implies that the equivariance between input space and a latent space is preserved for $G_S \cap G_L$ if the latent vector $\boldsymbol{z}$ is transformed by ψ_{G_S}. ■

We show that ψ_{G_S} preserves equivariance between G_L^a and G_I^a. If there exists equivariant function between input and latent vector space, there should be a group G_L for a latent space and its corresponding group G_I in an input space by definition of equivariance ($q_\phi(g_I x) = g_L q_\phi(x)$).

In other words, $\psi_{G_S}(\cdot)$ guarantees to preserve the equivariance of I2L-transformation to certain symmetries in $G_S \cap G_L$ after IPE-transformation as shown in Figure 2.

Let $P(B)$ be the probability of $\psi(\cdot) \in B$ for a subset $B \subset M_n(\mathbb{R})$ after VAE training, and $Pr(\psi_B \in B')$ be the conditional probability of $\psi(\cdot) \in B'$ given $\psi(\cdot) \in B$.

Then,

Proposition 4.3. $Pr(\psi_{E_S}(\cdot) \in G_S) > Pr(\psi_{E_M}(\cdot) \in G_S) > Pr(\psi_M(\cdot) \in G_S)$.

Proof. All $\mathbf{e}^{\boldsymbol{S}} \in E_S$ are in E_M since $Sym_n(\mathbb{R}) \subset M_n(\mathbb{R})$. However, $E_M \not\subset E_S$ because $\mathbf{e}^{\boldsymbol{S}}$ is always symmetric, but $\mathbf{e}^{\boldsymbol{M}}$ can be an asymmetric matrix. All elements of E_S are symmetric because of the matrix exponential property that $\mathbf{e}^{\boldsymbol{M}^\intercal} = (\mathbf{e}^{\boldsymbol{M}})^\intercal$. If $\boldsymbol{M}$ is a symmetric matrix then $\mathbf{e}^{\boldsymbol{M}^\intercal} = \mathbf{e}^{\boldsymbol{M}} = (\mathbf{e}^{\boldsymbol{M}})^\intercal$. Therefore, if $\boldsymbol{M}$ is symmetric then the exponential of $\boldsymbol{M}$ is also symmetric. We show a counter example to $E_M \subset E_S$. When $\boldsymbol{M} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$,

$$
\begin{aligned}
\mathbf{e}^{\boldsymbol{M}} &= \sum_{k=0}^{\infty} \frac{1}{k!} \boldsymbol{M}^k \\
&= I + \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} + \frac{1}{2!} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}^2 + \cdots + \frac{1}{(n-1)!} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}^{(n-1)} + \cdots \\
&= I + \begin{bmatrix} \sum_{n=0}^{\infty} \frac{1}{n!} & 1 + \sum_{n=0}^{\infty} \frac{1}{(n-1)!} \\ 0 & \sum_{n=0}^{\infty} \frac{1}{n!} \end{bmatrix} \\
&= \begin{bmatrix} 1+e & 1+e \\ 0 & 1+e \end{bmatrix}.
\end{aligned} \tag{6}
$$

The matrix $\mathbf{e}^{\boldsymbol{M}}$ is asymmetric and not in E_S. Therefore $E_M \not\subset E_S$. Therefore, the probability $Pr(\psi_{E_S}(\cdot) \in G_S) = \frac{P(G_S)}{P(E_S)}$ is greater than $Pr(\psi_{E_M}(\cdot) \in G_S) = \frac{P(G_S)}{P(E_M)}$. In the same way, $Pr(\psi_{E_M}(\cdot) \in G_S) > Pr(\psi_M(\cdot) \in G_S) = \frac{P(G_S)}{P(M_n(\mathbb{R}))}$ because $E_M \subset M_n(\mathbb{R})$ and non-invertible functions are only in $M_n(\mathbb{R})$. ■

Therefore, ψ_{E_S} clearly increases the probability of preserving a certain type of equivariance compared to unrestricted ψ functions.

The conditional probability $Pr(\psi_{E_S}(\cdot) \in G_S)$, $Pr(\psi_{E_M}(\cdot) \in G_S)$, and $Pr(\psi_M(\cdot) \in G_S)$ is changed by the distribution of the observation of $\psi(\cdot)$, which depends on the model parameters. However, the inequality $Pr(\psi_{E_S}(\cdot) \in G_S) > Pr(\psi_{E_M}(\cdot) \in G_S) > Pr(\psi_M(\cdot) \in G_S)$ is not changed regardless of the distribution of observation of $\psi(\cdot)$. We empirically validate the impact of equivariance with the uncertain $P(\cdot)$ to disentanglement in Section 6.1.5.

4.1.4 Relation Between $\psi(\cdot)$ and Disentanglement

In addition to invertible and partial-equivariant properties, our IPE-transformation also guarantees zero Hessian matrix, which enhances disentanglement without any additional loss of [35]. Hessian matrix of the transformation $\nabla_{\boldsymbol{z}}^2 \psi(\boldsymbol{z}) = \nabla_{\boldsymbol{z}}(\nabla_{\boldsymbol{z}} \mathrm{e}^{\boldsymbol{M}} \boldsymbol{z}) = 0$ because of the irrelevance of $\boldsymbol{M}$ to $\boldsymbol{z}$. By this property, $\psi(\cdot)$ leads that independently factorizes each dimension [35], and it injects group theory based inductive bias simultaneously. This is because the group decomposition of $\boldsymbol{z}$ space $G = G_1 \times G_2 \times \cdots \times G_k$ corresponds to group decomposition of the transformed latent vector $\hat{\boldsymbol{z}}$ space $G' = G_1' \times G_2' \times \cdots \times G_k'$ such that each G_i' is fixed by the action of all the G_j for $j \neq i$ [16], [36]. This correspondence of decomposition is expected to transfer the independence between dimensions of $\boldsymbol{z}$ to the space of $\hat{\boldsymbol{z}}$ [17].

Algorithm 1 Unit invertible and partial-equivariant Transformation Function (UIPET-function)

Require: matrices $\boldsymbol{M}_1$, and $\boldsymbol{M}_2$
Ensure: invertible and partial-equivariant Transformation Function $\psi(\cdot)$
$\boldsymbol{M}_1, \boldsymbol{M}_2 \leftarrow \frac{1}{2}(\boldsymbol{M}_1 + \boldsymbol{M}_1^\intercal), \frac{1}{2}(\boldsymbol{M}_2 + \boldsymbol{M}_2^\intercal)$
$\psi(\cdot) \leftarrow \boldsymbol{M}_1^\intercal \boldsymbol{M}_2$

Algorithm 2 IE-Transformation

Require: latent vector $\boldsymbol{z}$, and samples from prior $\boldsymbol{\epsilon}$
Ensure: transformed latent vector $\hat{\boldsymbol{z}}$, and transformed normal Guassian distribution samples $\hat{\boldsymbol{\epsilon}}$
$\psi(\cdot) \leftarrow$ UIET-function ($\boldsymbol{M}_1$, $\boldsymbol{M}_2$)
$\hat{\boldsymbol{z}}, \hat{\boldsymbol{\epsilon}} \leftarrow \psi(\boldsymbol{z}), \psi(\boldsymbol{\epsilon})$

Algorithm 3 KL Divergence & Posterior Estimator

Require: latent vector $\hat{\boldsymbol{z}}_m$, prior samples $\hat{\boldsymbol{\epsilon}}_m$,
Natural Parameter Generator $\Omega_1(\cdot), \Omega_2(\cdot)$
log-normalizer A, *sufficient statistics* T, and *evidence* $\boldsymbol{\nu}$.
Ensure: KL divergence $D_{\mathrm{KL}}(f_{\hat{\boldsymbol{z}}}(\hat{z}|\theta_{\hat{\boldsymbol{z}}})||f_{\hat{\epsilon}}(\hat{\epsilon}|\theta_{\hat{\epsilon}}))$, and posterior $p(\boldsymbol{\theta}|\mathbf{X}, \mathcal{X}, \boldsymbol{\nu})$
$\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}, \boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}} \leftarrow \Omega_1(\hat{\boldsymbol{z}}), \Omega_2(\hat{\boldsymbol{\epsilon}})$
$A \leftarrow$ implicit semantic mask(A) {Equation 22}
$p(\boldsymbol{\theta}|\mathbf{X}, \mathcal{X}, \boldsymbol{\nu}) \leftarrow \exp[\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}(\sum_{i=1}^{B} T(\hat{\boldsymbol{z}}_i) + \boldsymbol{\nu}\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}) - A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})]$ {Equation 8}
$D_{\mathrm{KL}}(f_{\hat{\boldsymbol{z}}}(\hat{\boldsymbol{z}}|\theta_{\hat{\boldsymbol{z}}})||f_{\hat{\boldsymbol{\epsilon}}}(\hat{\boldsymbol{\epsilon}}|\theta_{\hat{\boldsymbol{\epsilon}}})) \leftarrow A(\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}) - A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) + \boldsymbol{\theta}_{\hat{\boldsymbol{z}}}^\intercal \frac{\partial A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})}{\partial \boldsymbol{\theta}_{\hat{\boldsymbol{z}}}} - \boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}^\intercal \frac{\partial A(\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})}{\partial \boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}}$ {Equation 12}

Algorithm 4 EF-Conversion Loss

Require: KL divergence $D_{\mathrm{KL}}(f_{\hat{\boldsymbol{z}}}(\hat{z}|\theta_{\hat{\boldsymbol{z}}})||f_{\hat{\epsilon}}(\hat{\epsilon}|\theta_{\hat{\epsilon}}))$, posterior $p(\boldsymbol{\theta}|\mathbf{X},\mathcal{X},\boldsymbol{\nu})$, μ,σ
Ensure: Regularization $\mathcal{L}_{reg}$
$\mathcal{L}_{el} \leftarrow p(\boldsymbol{\theta}|\mathbf{X},\mathcal{X},\boldsymbol{\nu}) + \boldsymbol{\lambda}_m D_{\mathrm{KL}}(f_{\hat{\boldsymbol{z}}}(\hat{z}|\theta_{\hat{\boldsymbol{z}}})||f_{\hat{\epsilon}}(\hat{\epsilon}|\theta_{\hat{\epsilon}}))$
$\mathcal{L}_{el} \leftarrow ||\nabla_{\hat{\boldsymbol{z}}_m,\hat{\boldsymbol{\epsilon}}_m,\boldsymbol{\lambda}_m}\mathcal{L}_{el}||_2^2$
$D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})) \leftarrow 0.5\sum_{d=1}^{D}(1+2\log\sigma_j-\mu_j^2-\sigma_j^2)$ [5]
$\mathcal{L}_{cali} \leftarrow \mathrm{MSE}(D_{\mathrm{KL}}(f_{\hat{\boldsymbol{z}}}(\hat{z}|\theta_{\hat{\boldsymbol{z}}})||f_{\hat{\epsilon}}(\hat{\epsilon}|\theta_{\hat{\epsilon}})), D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})))$ {Equation 21}
$\mathcal{L} \leftarrow \mathcal{L}_{el} + \mathcal{L}_{cali}$ {Equation 33}

4.2 Exponential Family Conversion for Unknown Prior

In VAE frameworks, the Gaussian normal distribution is applied as a prior. However, a prior from data is usually unknown and may not follow the Gaussian distribution [14]. As a solution, we present a training procedure for VAEs to build an exponential family distribution from a latent variable of an arbitrary distribution. Then, we introduce training losses obtained from the unit IPE-transformation function and EF-conversion.

4.2.1 Elements of Exponential Family Distribution Settings

First, we set sufficient statistics $T(\cdot)$, log-normalizer $A(\cdot)$, and carrier or base measure $B(\cdot)$ are deterministic functions by maximizing conjugate prior for parameter ξ. To determine the *natural parameter* of posterior and prior $\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m}$, and $\hat{\epsilon}_m$, we use a natural parameter generator (NPG) designed by multi-layer perceptron [37]. As introduced in [37], [38], we assume exponential family always admits a conjugate prior:

$$q(\boldsymbol{\theta}|\xi,\boldsymbol{\nu}) = \exp(\boldsymbol{\nu}\boldsymbol{\theta}^{\intercal}\xi - \boldsymbol{\nu}A(\boldsymbol{\theta}) + B'(\xi,\boldsymbol{\nu})), \tag{7}$$

where $B'(\cdot)$ is a *normalize coefficient* and ν is evidence, and it is expressed by prior natural parameter ξ. However, generated natural parameter $\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m}$ is not guaranteed as the appropriate parameter of the exponential family corresponds to conjugate prior. To satisfy this condition, we assume observation is a set of independent identically distributed, then Eq. 2 is modified: $p(\mathbf{X}|\boldsymbol{\theta}) = \prod_{n=1}^{N} h(\mathbf{x}_n)\exp(\boldsymbol{\theta}^{\intercal}\sum_{n=1}^{N}T(\mathbf{x}_n) - A(\boldsymbol{\theta}))$ [38], where observation $\mathbf{X} = \{\mathbf{x}_1,\cdots\mathbf{x}_N\}$. In the next, we multiply the modified formation by the prior Eq. 7 to obtain the posterior distribution [38] as Eq. 8.

4.2.2 Distribution Approximation As an Exponential Family

The procedure represents a posterior distribution in the exponential family by adopting the following form:

$$p(\boldsymbol{\theta}|\mathbf{X},\xi,\boldsymbol{\nu}) \propto \exp(\boldsymbol{\theta}^{\intercal}(\sum_{n=1}^{N}T(\mathbf{x}_n) + \boldsymbol{\nu}\xi) - A(\boldsymbol{\theta})), \tag{8}$$

where *sufficient statistics* $T(\cdot)$ and *log-normalizer*, $A(\cdot)$ are known functions, samples $\mathbf{X} = \{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n\}$ from distribution, and *natural parameter* of posterior $\boldsymbol{\theta}$ and of prior ξ [38]. The functions $T(\cdot)$, and $A(\cdot)$ are deterministic functions to maximize posterior distribution. The *evidence* is implemented as learnable parameters $\boldsymbol{\nu} \in \mathbb{R}^{n\times n}$. The natural parameter is generated by a multi-layer perceptron as [37]. This general form approximating an exponential family distribution with learnable parameters can extend VAEs to use a wider distribution for latent variables by simply matching $\mathbf{X}$ to generated latent variables. After IPE-transformation, we can apply the form by using the $\hat{z}_m$, $\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m}$, and $\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}_m}$ for $\mathbf{X}$, $\boldsymbol{\theta}$, and ξ, respectively.

4.2.3 EF Similarity Loss

We added a loss to converge the unrestricted distributions of $\hat{z}$ to the power density function of the exponential family by constraining the posterior maximization as:

$$\begin{aligned}&\text{maximize } \log p(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m}|\hat{z}_m,\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}_m},\boldsymbol{\nu}_m)\\ &\qquad\text{s.t. } D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m})||f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}_m})) \geq 0\end{aligned} \tag{9}$$

$$\begin{aligned}\Rightarrow \mathcal{L}_s(\hat{z}_m,\hat{\epsilon}_m) = &\log p(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m}|\hat{z}_m,\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}_m},\boldsymbol{\nu}_m) +\\ &\boldsymbol{\lambda}_m D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m})||f_{\mathbf{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}_m}))\end{aligned} \tag{10}$$

$$\Rightarrow \mathcal{L}_{el} \coloneqq ||\nabla_{\hat{\boldsymbol{z}}_m,\hat{\boldsymbol{\epsilon}}_m,\boldsymbol{\lambda}_m}\mathcal{L}_s||_2^2 = 0. \tag{11}$$

This provides EF similarity loss in Eq. 33. The notation $\boldsymbol{\theta}_k$ is a generated natural parameter by a given $k \in \{\hat{z},\hat{\boldsymbol{\theta}}\}$, and $f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta})$ is a power density function of the exponential family. Moreover, $\boldsymbol{\lambda}_m$ is a trainable parameter for optimizing the Lagrange multiplier, and $D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m})||f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}_m}))$ is a KL divergence of the exponential family.

4.2.4 KL Divergence for Evidence of Lower Bound

The KL divergence of Gaussian distribution [5] is computed using mean and variance, which are the parameters of a Gaussian distribution. To introduce a loss as the KL divergence of Gaussian distribution, we compute KL divergence of the exponential family in Eq. 8 using the learnable parameter $T(\cdot)$ and $A(\cdot)$ with given natural parameter $\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}$ and $\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}$, expressed as:

$$\begin{aligned}\mathcal{L}_{kl} &\coloneqq D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m})||f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}_m}))\\ &= A(\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}) - A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) + \boldsymbol{\theta}_{\hat{\boldsymbol{z}}}^{\intercal}\nabla_{\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}}A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) - \boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}^{\intercal}\nabla_{\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}}A(\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}).\end{aligned} \tag{12}$$

Because $D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})||f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}))$ is followed as:

$$\begin{aligned}D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})||f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})) = &\int_{-\infty}^{\infty} f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\log f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\mathrm{d}\boldsymbol{x}\\ &-\int_{-\infty}^{\infty} f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\log f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})\mathrm{d}\boldsymbol{x}.\end{aligned} \tag{13}$$

We designed sufficient statistics as matrix multiplication (multi-layer perceptron). Then,

$$\begin{aligned}\int_{-\infty}^{\infty} f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\log f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\mathrm{d}\boldsymbol{x} &= \int_{-\infty}^{\infty} f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\cdot\\ &\quad[\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}^{\intercal}\mathbf{T}(\boldsymbol{x}) - A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) + B(\boldsymbol{x})]\mathrm{d}\boldsymbol{x}\\ &= -A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\int_{-\infty}^{\infty} f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\mathrm{d}\boldsymbol{x}\\ &\quad+\int_{-\infty}^{\infty} f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})[\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}^{\intercal}\mathbf{T}(\boldsymbol{x}) + B(\boldsymbol{x})]\mathrm{d}\boldsymbol{x}\\ &= -A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) + \boldsymbol{\theta}_{\hat{\boldsymbol{z}}}^{\intercal}\int_{-\infty}^{\infty} T(\boldsymbol{x})f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\mathrm{d}\boldsymbol{x}\\ &\quad+\int_{-\infty}^{\infty} B(\boldsymbol{x})f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\mathrm{d}\boldsymbol{x},\end{aligned} \tag{14}$$

and

$$\begin{aligned}\int_{-\infty}^{\infty} f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\log f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})\mathrm{d}\boldsymbol{x} &= -A(\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})\\ &\quad+\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}^{\intercal}\int_{-\infty}^{\infty} T(\boldsymbol{x})f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})\mathrm{d}\boldsymbol{x}\\ &\quad+\int_{-\infty}^{\infty} B(\boldsymbol{x})f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})\mathrm{d}\boldsymbol{x}.\end{aligned} \tag{15}$$

$$\begin{aligned}\therefore D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})||f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\epsilon}})) = A(\boldsymbol{\theta}_{\hat{\epsilon}}) - A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) \\ + \boldsymbol{\theta}_{\hat{\boldsymbol{z}}} \int_{-\infty}^{\infty} T(\boldsymbol{x}) f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) \mathrm{d}\boldsymbol{x} \\ - \boldsymbol{\theta}_{\hat{\epsilon}} \int_{-\infty}^{\infty} T(\boldsymbol{x}) f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\epsilon}}) \mathrm{d}\boldsymbol{x}.\end{aligned} \tag{16}$$

The mean of the sufficient statistic is followed as:

$$\int_{-\infty}^{\infty} T(\boldsymbol{x}) f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}) \mathrm{d}\boldsymbol{x} = \frac{\partial A^*(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \approx \frac{\partial A(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \quad \because A^*(\boldsymbol{\theta}) = \boldsymbol{\theta}^\intercal A^*, \tag{17}$$

where $A^*(\cdot)$ is a true log-partition function of the exponential family (ideal case of $A(\cdot)$). However, estimating A^* is difficult, and there is no direct method without random samplings, such as mini-batch weighted sampling or mini-batch stratified sampling [7]. Then, we approximate A^* to A, and train A to be close to A^*. Consequently, we obtain KL divergence of the exponential family as:

$$\begin{aligned}\int_{-\infty}^{\infty} f_{\boldsymbol{x}}(\mathrm{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) \log f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) = -A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) + \boldsymbol{\theta}_{\hat{\boldsymbol{z}}}^\intercal \frac{\partial A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})}{\partial \boldsymbol{\theta}_{\hat{\boldsymbol{z}}}} \\ + \int_{-\infty}^{\infty} f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) B(\boldsymbol{x}) \mathrm{d}\boldsymbol{x},\end{aligned} \tag{18}$$

$$\begin{aligned}\int_{-\infty}^{\infty} f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) \log f_{\boldsymbol{x}}(\mathrm{x}|\boldsymbol{\theta}_{\hat{\epsilon}}) = -Z(\boldsymbol{\theta}_{\hat{\epsilon}}) + \boldsymbol{\theta}_{\hat{\epsilon}}^\intercal \frac{\partial A(\boldsymbol{\theta}_{\hat{\epsilon}})}{\partial \boldsymbol{\theta}_{\hat{\epsilon}}} \\ + \int_{-\infty}^{\infty} f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) B(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}.\end{aligned} \tag{19}$$

Therefore, the final Kullback-Leibler divergence of exponential family is followed as:

$$D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})||f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\epsilon}})) = A(\boldsymbol{\theta}_{\hat{\epsilon}}) - A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) + \boldsymbol{\theta}_{\hat{\boldsymbol{z}}}^\intercal \frac{\partial A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})}{\partial \boldsymbol{\theta}_{\hat{\boldsymbol{z}}}} - \boldsymbol{\theta}_{\hat{\epsilon}}^\intercal \frac{\partial A(\boldsymbol{\theta}_{\hat{\epsilon}})}{\partial \boldsymbol{\theta}_{\hat{\epsilon}}}. \tag{20}$$

4.2.5 KL Divergence Calibration Loss

To reduce the error between the approximation and true matrix for the matrix exponential [39], we add a loss to minimize the difference of their KL divergence measured by mean squared error (MSE) as:

$$\begin{aligned}\mathcal{L}_{cali} = \mathrm{MSE}(D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z})), \\ D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m})||f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\epsilon}_m}))),\end{aligned} \tag{21}$$

which is the KL divergence calibration loss ($\mathcal{L}_{cali}$).

4.2.6 Implicit Semantic Mask

We propose an implicit semantic mask to improve disentanglement learning. We apply mask matrix $\mathcal{M}$ which consists of 0 or 1 element to log-normalizer to prevent less effective weight flow as:

$$\mathcal{M}_{ij} = \begin{cases} 1 \text{ if } |\mathcal{W}_{ij}| \geq \mu_{|\mathcal{W}_{ij}|} - \lambda\sigma_{|\mathcal{W}_{ij}|} \\ 0 \text{ otherwise} \end{cases}, \tag{22}$$

where $\mathcal{W}$ is the weight of log-normalizer, λ is a hyper-parameter, $\mu_{|\mathcal{W}_{ij}|}$, and $\sigma_{|\mathcal{W}_{ij}|}$ are the mean, and standard deviation of weight respectively. Previous work [19] utilizes a semantic mask in input space directly, but we inject the semantic mask implicitly on the latent space.

TABLE 2: VAE architecture for dSprites dataset.

Encoder	Decoder
Input 64 × 64 binary image	input ∈ $\mathbb{R}^{10}$
4 × 4 conv. 32 ReLU. stride 2	FC. 128 ReLU.
4 × 4 conv. 32 ReLU. stride 2	FC. 4 × 4 × 64 ReLU.
4 × 4 conv. 64 ReLU. stride 2	4 × 4 upconv. 64 ReLU. stride 2.
4 × 4 conv. 64 ReLU. stride 2	4 × 4 upconv. 32 ReLU. stride 2.
FC. 128. FC. 2 × 10	4 × 4 upconv. 32 ReLU. stride 2.
	4 × 4 upconv. 1. stride 2

TABLE 3: VAE architecture for 3D Shapes, and 3D Cars datasets. For exceptional case, CLG-VAE, we ues ten dimension size on 3D Shapes dataset [15].

Encoder	Decoder
Input 64 × 64 × 3 RGB image	input ∈ $\mathbb{R}^6$ (3D Shapes), $\mathbb{R}^{10}$ (3D Cars)
4 × 4 conv. 32 ReLU. stride 2	FC. 256 ReLU.
4 × 4 conv. 32 ReLU. stride 2	FC. 4 × 4 × 64 ReLU.
4 × 4 conv. 64 ReLU. stride 2	4 × 4 upconv. 64 ReLU. stride 2.
4 × 4 conv. 64 ReLU. stride 2	4 × 4 upconv. 32 ReLU. stride 2.
FC. 256. FC. 2 × 10	4 × 4 upconv. 32 ReLU. stride 2.
	4 × 4 upconv. 3. stride 2

TABLE 4: dSprites and 3D Cars: epochs for dSprites and 3D cars are 30 and 200, respectively.

models	hyper-parameters	values
common	batch size	256
	epoch	{30, 200}
	optim	Adam
	lr	4e-4
	lr for MIPET	4e-4
	weight decay	1e-4
	latent dim	10
β-VAE	# of IE and EF	{1, 2, 4, 10}
β-TCVAE	β	{4, 6}
	# of IE and EF	{1, 3}
	α, γ	1.0
CLG-VAE	λ_{decomp}	40
	λ_{hessian}	40
	forward group	0.2
	group reconst	{0.2, 0.5, 0.7}

TABLE 5: lr is learning rate, latent dim is dimension size of latent vector, group reconst is group reconstrunction, and forward group is forward group pass.

models	hyper-parameters	values
common	batch size	256
	epoch	67
	optim	Adam
	lr	4e-4
	lr for MIPET	4e-4
β-VAE	# of IE and EF	{1, 2, 4, 10}
	weight decay	0.0
	latent dim	6
β-TCVAE	β	{4, 6}
	# of IE and EF	{1, 3}
	α, γ	1.0
	weight decay	1e-4
	latent dim	6
CLG-VAE	λ_{decomp}	40
	λ_{hessian}	40
	forward group	0.2
	group reconst	{0.2, 0.5, 0.7}
	weight decay	0.0
	latent dim	10

4.3 Integration for Multiple IPE-Transformation and EF-Conversion

We mathematically extend IPE-transformation to MIPE-transformation, which is the equivalent process of β-VAE to enhance disentanglement. Each IPE-transformation function operates independently, then the reconstruction error for objective function is defined as:

$$\mathcal{L}_{rec} := \frac{1}{k}\sum_{i=1}^{k}\left[\int q_i(\hat{\boldsymbol{z}}_i|\boldsymbol{x})\log p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_i)\mathrm{d}\hat{\boldsymbol{z}}_i \prod_{j=1,j\neq i}^{k}\int q_j(\hat{\boldsymbol{z}}_j|\boldsymbol{x})\mathrm{d}\hat{\boldsymbol{z}}_j\right] = \frac{1}{k}\sum_{i=1}^{k}E_{q_{\phi,\psi_i}(\boldsymbol{z}|\boldsymbol{x})}\log p_\theta(\boldsymbol{x}|\psi_i(\boldsymbol{z})), \tag{23}$$

where $\hat{\boldsymbol{z}}_i = \psi_i(\boldsymbol{z})$. Becuase the log likelihood of $p(\boldsymbol{x})$ can be derived as follows:

$$\log p_\theta(\boldsymbol{x}) = \int\prod_i^k q_1(\hat{\boldsymbol{z}}_i|\boldsymbol{x})\log p_\theta(\boldsymbol{x})\mathrm{d}\hat{\boldsymbol{z}}' \tag{24}$$

$$= \int\prod_i^k q_1(\hat{\boldsymbol{z}}_i|\boldsymbol{x})\log\frac{p_\theta(\boldsymbol{x},\hat{\boldsymbol{z}}_1,\hat{\boldsymbol{z}}_2,\cdots,\hat{\boldsymbol{z}}_k)}{p_\theta(\hat{\boldsymbol{z}}_1,\hat{\boldsymbol{z}}_2,\cdots,\hat{\boldsymbol{z}}_k|\boldsymbol{x})}\mathrm{d}\hat{\boldsymbol{z}}' \tag{25}$$

$$= \int\prod_i^k q_1(\hat{\boldsymbol{z}}_i|\boldsymbol{x})\cdot \left[\log\frac{p_\theta(\boldsymbol{x},\hat{\boldsymbol{z}}_1,\hat{\boldsymbol{z}}_2,\cdots,\hat{\boldsymbol{z}}_k)}{q(\hat{\boldsymbol{z}}_1,\hat{\boldsymbol{z}}_2,\cdots,\hat{\boldsymbol{z}}_k|\boldsymbol{x})} - \log\frac{p_\theta(\hat{\boldsymbol{z}}_1,\hat{\boldsymbol{z}}_2,\cdots,\hat{\boldsymbol{z}}_k|\boldsymbol{x})}{q(\hat{\boldsymbol{z}}_1,\hat{\boldsymbol{z}}_2,\cdots,\hat{\boldsymbol{z}}_k|\boldsymbol{x})}\right]\mathrm{d}\hat{\boldsymbol{z}}' \tag{26}$$

$$\geqq \int\prod_i^k q_1(\hat{\boldsymbol{z}}_i|\boldsymbol{x})\log\frac{p_\theta(\boldsymbol{x},\hat{\boldsymbol{z}}_1,\hat{\boldsymbol{z}}_2,\cdots,\hat{\boldsymbol{z}}_k)}{q(\hat{\boldsymbol{z}}_1,\hat{\boldsymbol{z}}_2,\cdots,\hat{\boldsymbol{z}}_k|\boldsymbol{x})}\mathrm{d}\hat{\boldsymbol{z}}' \tag{27}$$

$$= \int\prod_i^k q_1(\hat{\boldsymbol{z}}_i|\boldsymbol{x})\cdot \left[\log p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_1,\hat{\boldsymbol{z}}_2,\cdots,\hat{\boldsymbol{z}}_k) + \log\frac{p(\hat{\boldsymbol{z}}_1,\hat{\boldsymbol{z}}_2,\cdots,\hat{\boldsymbol{z}}_k)}{q(\hat{\boldsymbol{z}}_1,\hat{\boldsymbol{z}}_2,\cdots,\hat{\boldsymbol{z}}_k|\boldsymbol{x})}\right]\mathrm{d}\hat{\boldsymbol{z}}', \tag{28}$$

where $\mathrm{d}\hat{\boldsymbol{z}}' = \mathrm{d}\hat{\boldsymbol{z}}_1\mathrm{d}\hat{\boldsymbol{z}}_2\cdots\mathrm{d}\hat{\boldsymbol{z}}_k$. Each IPE-transformation function operates independently, then $\log p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_1,\hat{\boldsymbol{z}}_2,\cdots,\hat{\boldsymbol{z}}_k) = -(k-1)\log p_\theta(\boldsymbol{x}) + \Pi_{i=1}^{k}p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_i)$. Then,

$$\begin{aligned} p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_1,\hat{\boldsymbol{z}}_2,\ldots,\hat{\boldsymbol{z}}_k) &= \frac{p_\theta(\hat{\boldsymbol{z}}_1,\hat{\boldsymbol{z}}_2,\ldots,\hat{\boldsymbol{z}}_k|\boldsymbol{x})p_\theta(\boldsymbol{x})}{p_\theta(\hat{\boldsymbol{z}}_1,\hat{\boldsymbol{z}}_2,\ldots,\hat{\boldsymbol{z}}_k)} \\ &= \frac{p_\theta(\boldsymbol{x})\prod_{i=1}^{k}p_\theta(\hat{\boldsymbol{z}}_i|\boldsymbol{x})}{\prod_{i=1}^{k}p_\theta(\hat{\boldsymbol{z}}_i)}(\because(\hat{\boldsymbol{z}}_i \perp\!\!\!\perp \hat{\boldsymbol{z}}_j s|\boldsymbol{x})) \\ &= \prod_{i=1}^{k}\frac{p_\theta(\hat{\boldsymbol{z}}_i|\boldsymbol{x})p_\theta(\boldsymbol{x}^{\frac{1}{k}})}{p_\theta(\hat{\boldsymbol{z}}_i)} \\ &= p_\theta(\boldsymbol{x})^{-(k-1)}\prod_{i=1}^{k}\frac{p_\theta(\hat{\boldsymbol{z}}_i|\boldsymbol{x})p_\theta(\boldsymbol{x})}{p_\theta(\hat{\boldsymbol{z}}_i)} \\ &= p_\theta(\boldsymbol{x})^{-(k-1)}\prod_{i=1}^{k}p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_i), \end{aligned} \tag{29}$$

where $\hat{\boldsymbol{z}}_j s = \cap_{j=1,j\neq i}^{k}\hat{\boldsymbol{z}}_j$. Therefore,

$$\begin{aligned} &\int\prod_{i=1}^{k}q_i(\hat{\boldsymbol{z}}_i|\boldsymbol{x})\log p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_1,\hat{\boldsymbol{z}}_2,\cdots,\hat{\boldsymbol{z}}_k)\mathrm{d}\hat{\boldsymbol{z}}' \\ &= \int\prod_{i=1}^{k}q_i(\hat{\boldsymbol{z}}_i|\boldsymbol{x})\Big[-(k-1)\log p_\theta(\boldsymbol{x}) + \prod_{i=1}^{k}p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_i)\Big]\mathrm{d}\hat{\boldsymbol{z}}' \\ &= -(k-1)\log p_\theta(\boldsymbol{x}) + \int\prod_{i=1}^{k}q_i(\hat{\boldsymbol{z}}_i|\boldsymbol{x})\prod_{i=j}^{k}p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_j)\mathrm{d}\hat{\boldsymbol{z}}'. \end{aligned} \tag{30}$$

Then,

$$\begin{aligned} \log p_\theta(\boldsymbol{x}) \geq{}& \frac{1}{k}\sum_{i=1}^{k}\left[\int q_i(\hat{\boldsymbol{z}}_i|\boldsymbol{x})\log p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_i)\mathrm{d}\hat{\boldsymbol{z}}_i\prod_{j=1,j\neq i}^{k}\int q_j(\hat{\boldsymbol{z}}_j|\boldsymbol{x})\mathrm{d}\hat{\boldsymbol{z}}_j\right] \\ &- \int\prod_i^k q_1(\hat{\boldsymbol{z}}_i|\boldsymbol{x})\log\frac{\prod_{i=1}^{k}q_i(\hat{\boldsymbol{z}}_i|\boldsymbol{x})}{\prod_{i=1}^{k}p(\hat{\boldsymbol{z}}_i)}\mathrm{d}\hat{\boldsymbol{z}}' \\ ={}& \frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_{q(\hat{\boldsymbol{z}}_i|\boldsymbol{x})}\log p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_i) \\ &- \sum_{i=1}^{k}\Big[D_{\mathrm{KL}}(q_\phi(\hat{\boldsymbol{z}}_i|\boldsymbol{x})||p(\hat{\boldsymbol{z}}_i))\prod_{j=1,j\neq i}^{k}\int q_j(\hat{\boldsymbol{z}}_j|\boldsymbol{x})\mathrm{d}\hat{\boldsymbol{z}}_j\Big] \\ ={}& \frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_{q_\phi(\hat{\boldsymbol{z}}_i|\boldsymbol{x})}\log p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_i) - \sum_{i=1}^{k}D_{\mathrm{KL}}(q_\phi(\hat{\boldsymbol{z}}_i|\boldsymbol{x})||p(\boldsymbol{z}_i)) \\ ={}& \frac{1}{k}\Big[\sum_{i=1}^{k}\mathbb{E}_{q_\phi(\hat{\boldsymbol{z}}_i|\boldsymbol{x})}\log p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_i) - kD_{\mathrm{KL}}(q_\phi(\hat{\boldsymbol{z}}_i|\boldsymbol{x})||p(\hat{\boldsymbol{z}}_i))\Big]. \end{aligned} \tag{31}$$

Therefore, we define ELBO as:

$$\mathcal{L}'(\phi,\theta,\psi_{i\in[1,k]};\boldsymbol{x}) = \frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_{q_{\phi,\psi_i}(\boldsymbol{z}_i|\boldsymbol{x})}\log p_\theta(\boldsymbol{x}|\psi_i(\boldsymbol{z})) - \sum_{i=1}^{k}D_{\mathrm{KL}}(q_{\phi,\psi_i}(\boldsymbol{z}|\boldsymbol{x})||p_{\psi_i}(\boldsymbol{z})). \tag{32}$$

However, following Eq. 32, k samples are generated, and each sample is disentangled for different factors. We implement the output as the average of the sum of the k samples to obtain a single sample with a superposition effect from k samples. Moreover, the KL divergence term in Eq. 32 represents that increasing number of MIPE-transformation is equal to increasing β hyper-parameter in β-VAE [6].

The VAEs equipped with MIPE-transformation (MIPET-VAEs) can be trained with the following loss:

$$\mathcal{L}(\phi,\theta,\psi_{i\in[1,k]};\boldsymbol{x}) = \mathcal{L}_{rec} - \mathcal{L}_{kl} - \mathcal{L}_{el} - \mathcal{L}_{cali}. \tag{33}$$

The whole process to define objective function is represented in Algorithm 1-5.

5 Experiment Settings

5.0.1 Models

As baseline models, we select VAE, β-VAE, β-TCVAE, and CLG-VAE. These models are compared to their extension to adopt MIPET, abbreviated by adding the MIPET prefix. We apply the proposed method to β-TCVAE only with the EF similarity loss term because β-TCVAE penalizes the divided KL divergence terms. We set the same encoder and decoder architecture in each model to exclude the overlapped effects. Also, we follow the same model architecture which are introduced in previous works [8] and model details are in Table 2-3.

5.0.2 Datasets

We compare well-known VAEs to CHIC-VAEs: VAE, β-VAE, β-TCVAE, and CLG-VAE on the following data sets with 1) dSprites [40] which consists of 737,280 binary 64×64 images of dSprites with five independent ground truth factors(number of values), *i.e.* shape(3), orientation(40), scale(6), x-position(32), and y-position(32). 2) 3D Shapes [41] which consists of 480,000 RGB $64 \times 64 \times 3$ images of 3D Shapes with six independent ground truth factors: shape(4) orientation(15), scale(8), wall color(10), floor color(10), and

TABLE 6: Performance (mean $\pm$ std) of four metrics on dSprites, 3D Shapes, and 3D Cars. The $\alpha = 1$ and $\gamma = 1$ of β-TCVAE as [7].

dSprites	Disentanglement Metric							
	FVM ↑		MIG ↑		SAP ↑		DCI ↑	
	original	MIPET	original	MIPET	original	MIPET	original	MIPET
β-VAE	69.15($\pm$5.88)	**74.19**($\pm$5.62)	9.49($\pm$8.30)	**19.72**($\pm$11.37)	2.43($\pm$2.07)	**5.08**($\pm$2.90)	18.57($\pm$12.41)	**28.81**($\pm$10.19)
β-TCVAE	78.50($\pm$7.93)	**79.87**($\pm$5.80)	26.00($\pm$9.06)	**35.04**($\pm$4.07)	7.31($\pm$0.61)	**7.70**($\pm$1.63)	41.80($\pm$8.55)	**47.83**($\pm$5.01)
CLG-VAE	79.06($\pm$6.83)	**81.80**($\pm$3.17)	23.40($\pm$7.89)	**36.34**($\pm$5.55)	7.37($\pm$0.96)	**8.03**($\pm$0.83)	37.68($\pm$7.83)	**44.73**($\pm$5.11)
Control-VAE	62.36($\pm$8.62)	**67.71**($\pm$6.41)	4.36($\pm$2.86)	**7.34**($\pm$4.10)	**2.11**($\pm$1.88)	1.93($\pm$1.63)	10.40($\pm$3.42)	**15.18**($\pm$4.61)

3D Shapes	Disentanglement Metric							
	FVM ↑		MIG ↑		SAP ↑		DCI ↑	
	original	MIPET	original	MIPET	original	MIPET	original	MIPET
β-VAE	71.76($\pm$12.26)	**75.19**($\pm$8.16)	37.33($\pm$22.34)	**47.37**($\pm$10.13)	7.48($\pm$4.12)	**9.20**($\pm$2.44)	52.07($\pm$17.92)	**54.95**($\pm$8.99)
β-TCVAE	76.62($\pm$10.23)	**80.59**($\pm$8.57)	52.93($\pm$20.5)	**54.49**($\pm$9.44)	10.64($\pm$5.93)	**11.58**($\pm$3.32)	65.32($\pm$11.37)	**66.22**($\pm$7.32)
CLG-VAE	77.04($\pm$8.22)	**80.17**($\pm$8.43)	49.74($\pm$8.18)	**53.87**($\pm$7.41)	9.20($\pm$2.44)	**12.83**($\pm$3.01)	57.70($\pm$8.60)	**60.74**($\pm$7.77)
Control-VAE	71.05($\pm$14.35)	**71.89**($\pm$8.33)	24.88($\pm$13.68)	**32.28**($\pm$10.74)	6.60($\pm$3.59)	**7.14**($\pm$2.09)	40.08($\pm$13.45)	**43.06**($\pm$8.68)

3D Cars	Disentanglement Metric							
	FVM ↑		MIG ↑		SAP ↑		DCI ↑	
	original	MIPET	original	MIPET	original	MIPET	original	MIPET
β-VAE	**89.48**($\pm$5.22)	88.95($\pm$5.94)	6.90($\pm$2.70)	**7.27**($\pm$1.99)	1.30($\pm$0.48)	**1.88**($\pm$1.12)	**19.85**($\pm$4.87)	18.90($\pm$4.49)
β-TCVAE	95.84($\pm$3.40)	**96.43**($\pm$2.42)	**11.87**($\pm$2.90)	10.80($\pm$1.22)	1.55($\pm$0.38)	**1.88**($\pm$1.12)	**27.91**($\pm$4.31)	26.08($\pm$2.47)
CLG-VAE	86.11($\pm$7.12)	**91.06**($\pm$5.09)	6.19($\pm$2.42)	**8.51**($\pm$2.11)	**2.06**($\pm$0.60)	1.99($\pm$0.93)	16.91($\pm$4.01)	**18.31**($\pm$2.83)
Control-VAE	88.76($\pm$7.66)	**89.10**($\pm$6.90)	4.68($\pm$2.67)	**5.08**($\pm$2.68)	1.16($\pm$0.74)	**1.45**($\pm$0.86)	14.70($\pm$3.84)	**15.22**($\pm$4.15)

TABLE 7: p-value of t-test for original vs MIPET results of Table 6, which are averaged over models (bold: positive and significant, italic: positive but insignificant, normal: lower performance).

p-value	VAEs				CLG-VAE				β-TCVAEs			
	FVM	MIG	SAP	DCI	FVM	MIG	SAP	DCI	FVM	MIG	SAP	DCI
dSprites	**0.000**	**0.000**	**0.000**	**0.000**	**0.030**	**0.000**	**0.005**	**0.000**	*0.281*	**0.000**	*0.170*	**0.009**
3D Shapes	**0.080**	**0.007**	**0.016**	*0.191*	*0.085*	**0.029**	**0.000**	*0.088*	*0.111*	*0.383*	*0.277*	*0.390*
3D Cars	0.659	*0.250*	**0.003**	0.583	**0.003**	**0.000**	0.630	*0.071*	*0.278*	0.923	*0.119*	0.933

object color(10). 3) 3D Cars [42] which consists of 17,568 RGB $64 \times 64 \times 3$ images of 3D Shapes with three independent ground truth factors: car models(183), azimuth directions(24), and elevations(4).

5.0.3 Training

We set 256 mini-batch size in the datasets (dSprites, 3D Shapes, and 3D Cars), Adam optimizer with learning rate 4×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and epochs from $\{30, 67, 200\}$ as a common setting for all the comparative methods. For the comparison, we follow training and inference on the whole dataset. We train each model for 30, 67, and 200 epochs on the dSprites, 3D Shapes, and 3D Cars, respectively, as introduced in [8], [43]. We tune β from $\{1, 2, 4, 10\}$ and $\{4, 6\}$ for β-VAE and β-TCVAE, respectively. We set the dimension size of the latent vectors from $\{6, 10\}$ for 10 on dSprites and 3D Cars datasets and 6 for 3D Shapes, but we set 10 for CLG-VAE because it sets 10 dimensions size on 3D Shapes in [15]. Regarding the CLG-VAE, we fix λ_{decomp}, λ_{hessian}, and forward group features as 40, 20, and 0.2, respectively. Because the hyper-parameters showed the best result in [15]. We set group reconstruction from $\{0.2, 0.5, 0.7\}$. For Control-VAE, we set the maximum KL divergence value from $\{10, 12, \ldots, 20\}$. In addition, we set masking ratio λ from $\{0.0, 0.5, \ldots 2.0, \infty\}$. To check the impact of MIPE-transformation, we do not consider the Groupified VAE because the latter is implemented with an extended decoder (different capacity).

5.0.4 Evaluation

We conduct experiments on NVIDIA A100, RTX 2080 Ti, and RTX 3090. We set 100 samples to evaluate global empirical variance in each dimension and run it a total of 800 times to estimate the FVM score introduced in [8]. For the MIG [7], SAP [44], and DCI [23], we follow default values introduced in [45], training and evaluation 100 and 50 times with 100 mini-batches, respectively. We evaluate four disentanglement metrics for a less biased understanding of the actual states of disentanglement.

6 Results and Discussion

6.1 Quantitative Analysis

6.1.1 Disentanglement Metrics

We set the number of IPE-transformation functions to be equal to balancing hyper-parameter β on β-VAE because of Eq. 33. The number of IPE-transform functions of β-TCVAE is 3. However, in the case of CLG-VAE, we set it to 1 because its approach is based on the group theory, not directly controlling a KL divergence term such as β-VAE. We average each model performance value with 40, 20, 60, and 30 cases in VAEs, β-TCVAEs, Control-VAE and CLG-VAEs, respectively.

As shown in Table 6, MIPET-VAEs disentanglement performance is broadly improved with four metrics on each dataset. In particular, most FVM results significantly affect the model performance and stability on all datasets. Therefore, our proposed method obtains a specific dimension that corresponds to a specific single factor. These results imply that applied to MIPE-transformation functions on VAEs elaborate disentangled representation learning.

We additionally estimate the p-value of each metrics over models in Table 7. Previous work shows the average case of each models [16].

TABLE 8: Impact of the number of MIPE-transformation function on the β-TCVAE and β-VAE with dSprites, 3D Shapes, and 3D Cars datasets in terms of the four metrics. The blue and red box plots represent each model's single and multiple IPE-transformation cases, respectively. (A-n: MIPET-β-TCVAE (4), B-n: MIPET-β-TCVAE (6), C-n: MIPET-β-VAE, n: the number of MIPE-transformation).

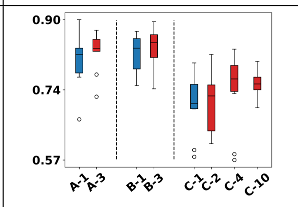
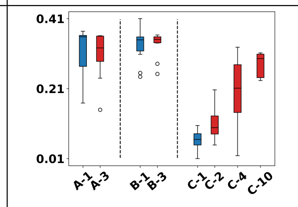
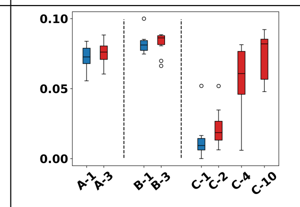
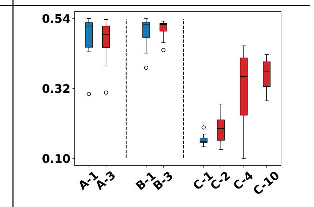
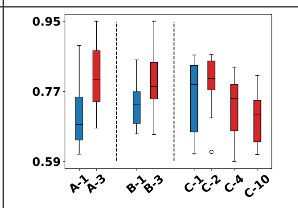
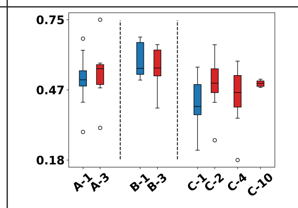
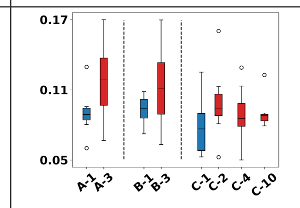
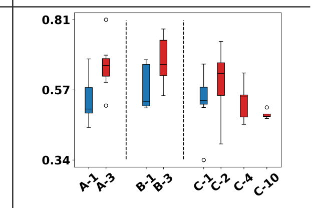
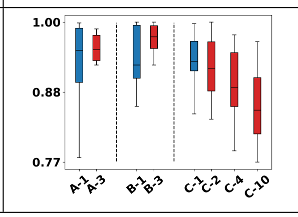
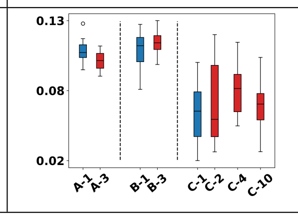
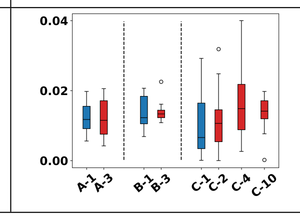
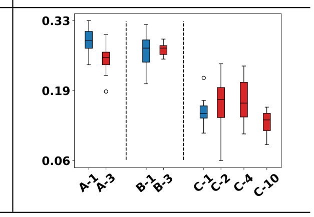
dataset	Metrics			
	FVM	IMG	SAP	DCI
dSprites				
3D Shapes				
3D Cars				

TABLE 9: Impact of the mask (mean$\pm$std.) and its ratio λ in Eq. 22 on 3D Cars. (∞: no masking case, gray box: the best setting over all metrics, bold text: the best in each metric.) Each model runs with ten random seeds.

ratio	β-VAE (1)				CLG-VAE (0.5)			
λ	FVM $\uparrow$	MIG $\uparrow$	SAP $\uparrow$	DCI $\uparrow$	FVM $\uparrow$	MIG $\uparrow$	SAP $\uparrow$	DIC $\uparrow$
0.0	90.46($\pm$6.50)	4.84($\pm$2.32)	1.29($\pm$0.81)	16.76($\pm$4.68)	**90.06**($\pm$4.44)	**9.28**($\pm$2.09)	1.82($\pm$0.82)	**19.12**($\pm$3.41)
0.5	91.35($\pm$5.52)	5.37($\pm$2.74)	1.17($\pm$0.67)	16.65($\pm$3.76)	88.69($\pm$4.78)	6.90($\pm$1.96)	1.85($\pm$0.67)	17.52($\pm$3.16)
1.0	**91.78**($\pm$6.20)	4.99($\pm$2.27)	1.36($\pm$0.81)	16.50($\pm$2.53)	83.60($\pm$11.48)	8.12($\pm$3.66)	**2.37**($\pm$1.50)	17.07($\pm$3.89)
1.5	90.04($\pm$5.88)	**7.22**($\pm$2.87)	**1.36**($\pm$0.48)	**18.23**($\pm$2.84)	84.76($\pm$6.86)	7.70($\pm$2.11)	2.05($\pm$0.73)	17.06($\pm$2.77)
2.0	87.79($\pm$8.88)	4.75($\pm$2.49)	1.01($\pm$0.99)	16.64($\pm$3.75)	85.78($\pm$4.18)	7.83($\pm$1.79)	1.91($\pm$0.96)	17.26($\pm$2.07)
∞	89.43($\pm$11.72)	3.74($\pm$2.32)	0.77($\pm$0.39)	15.45($\pm$4.59)	82.96($\pm$11.84)	8.07($\pm$2.52)	2.32($\pm$1.02)	17.46($\pm$4.07)

We divide each case into four categories: 1) Positive & Significant, 2) Positive & Insignificant, 3) Negative & Insignificant, and 4) Negative & Significant, where positive is when the mean value is higher than baseline and significant is statistically significant. We estimate the probability of each category: 1) 50%, 2) 36.11%, and 3) 13.89%. As shown in Table 7 and the results, half of the cases are statistically significant, and 86.11% of cases are improved model performance. Even though our method shows a lower value than the baseline, it is not significantly decreased (13.89%). In addition, averaged results show that our method impacts to model itself without hyper-parameter tuning. β-TCVAEs is partially using our method (paragraph Models in Section 5), so it does not show the whole effect of MIPET, but it improves model performance in many cases.

6.1.2 Sensitivity to the Number of IPE-transformation and EF-conversion

We investigate the impact of the MIPE-transformation function. As presented in Table. 8, MIPE-transformation is better than IPE-transformation for disentanglement learning on each dataset. Indeed, MIPET-β-VAEs results more generally and clearly show the impact of the MIPE-transfomation function. Our derivation in Section 4.3 clearly explains MIPE-transformation impact. This result shows the impact of the multiple uses of IPE-transformation and EF-conversion.

6.1.3 Impact of Implicit Semantic Mask

We set masking hyper-parameter λ from $\{0.0, 0.5, \cdots, 2.0, \infty\}$, and each model has different λ for best case. In Table 9, VAE and CLG-VAE with masked log-normalizer show better and well-balanced results than the models without masking, which implies improvement of disentanglement.

6.1.4 Ablation Study

We conduct an ablation study to evaluate the separate impact of equivariant property and the EF-conversion. We have already presented the impact of the multiple uses of IPE-transform and EF-conversion in the previous paragraph. We evaluate the impact of the other properties by setting MIPE-transformation 1) without equivariant (w/o E), which is implemented as an asymmetric matrix, and 2) without EF-conversion (w/o EF). To exclude group theory interference with other methods, we select β-VAE and β-TCVAE. As the results are shown in Table 10, most of the results show

TABLE 10: Ablation study for the equivariant property (w/o E), and EF-conversion (w/o EF). Each metric is averaged over 40 and 20 settings of β-VAE and β-TCVAE, respectively.

3D Cars	β-VAE			β-TCVAE		
	MIPET	MIPET (w/o E)	MIPET (w/o EF)	MIPET	MIPET (w/o E)	MIPET (w/o EF)
FVM ↑	**88.95**(±5.94)	82.09(±11.33)	45.23(±6.39)	**96.43**(±2.42)	91.34(±4.75)	91.43(±4.86)
MIG ↑	**7.27**(±1.99)	6.77(±2.41)	0.04(±0.02)	**10.80**(±1.22)	9.79(±1.07)	9.81(±1.10)
SAP ↑	**1.88**(±1.12)	1.76(±1.06)	0.18(±0.12)	**1.88**(±1.12)	1.35(±0.30)	1.35(±0.30)
DCI ↑	**18.90**(±4.49)	17.21(±5.57)	1.67(±1.26)	**26.08**(±2.47)	25.12(±3.72)	25.16(±3.82)
dSprites	β-VAE			β-TCVAE		
	MIPET	MIPET (w/o E)	MIPET (w/o EF)	MIPET	MIPET (w/o E)	MIPET (w/o EF)
FVM	**74.19**(±5.62)	71.54(±8.66)	25.83(±1.16)	**79.87**(±5.80)	76.39(±7.44)	77.44(±7.15)
MIG	**19.72**(±11.37)	19.29(±11.79)	0.02(±0.01)	**35.04**(±4.07)	33.83(±8.06)	21.88(±8.42)
SAP	**5.08**(±2.90)	4.91(±3.25)	0.21(±0.10)	**7.70**(±1.63)	7.64(±2.03)	6.84(±1.87)
DCI	**28.81**(±10.19)	27.51(±11.49)	1.81(±0.08)	**47.83**(±5.01)	45.10(±6.92)	37.84(±8.85)
3D Shapes	β-VAE			β-TCVAE		
	MIPET	MIPET (w/o E)	MIPET (w/o EF)	MIPET	MIPET (w/o E)	MIPET (w/o EF)
FVM	**75.19**(±8.16)	74.91(±10.46)	22.27(±1.29)	**80.59**(±8.57)	77.90(±8.66)	66.38(±7.57)
MIG	47.37(±10.13)	**47.45**(±8.98)	0.28(±0.09)	**54.49**(±9.44)	51.37(±11.54)	36.08(±17.42)
SAP	9.20(±2.44)	**9.43**(±2.59)	0.26(±0.07)	**11.58**(±3.32)	10.23(±3.13)	7.13(±3.09)
DCI	**54.95**(±8.99)	54.23(±9.05)	0.10(±0.02)	**66.22**(±7.32)	61.18(±8.87)	56.85(±11.72)

TABLE 11: The ratio of seeds to show better performance with symmetric matrix

dSprites	3D Shapes	3D Cars
0.58	0.56	0.67

TABLE 12: Training complexity.

# of IE	Complexity
0	× 1.00
1	× 0.75
3	× 0.50
4	× 0.33

that MIPET-VAEs performance is better than other cases. In particular, MIPET (w/o EF) results are lower than MIPET (w/o E) results and are clearly shown in all cases.

6.1.5 Impact of Symmetric Matrix Exponential

We empirically show the benefit of using a symmetric matrix for ψ. Table 11 shows the ratio of runs with a symmetric matrix, which shows better performance than unrestricted matrices, to the total 240 (60 models $\times$ 4 metrics) runs for each dataset. All results are higher than 0.5, which implies that the constraint enhances I2L equivariance even with uncertain factors.

6.1.6 Additional Experiment of Computing Complexity

We additionally estimate the computing complexity depending on the number of IPE-transformation. The results are in Table 12 and represent the training time complexity compare to baselines (when the number of IE is equal to 0).

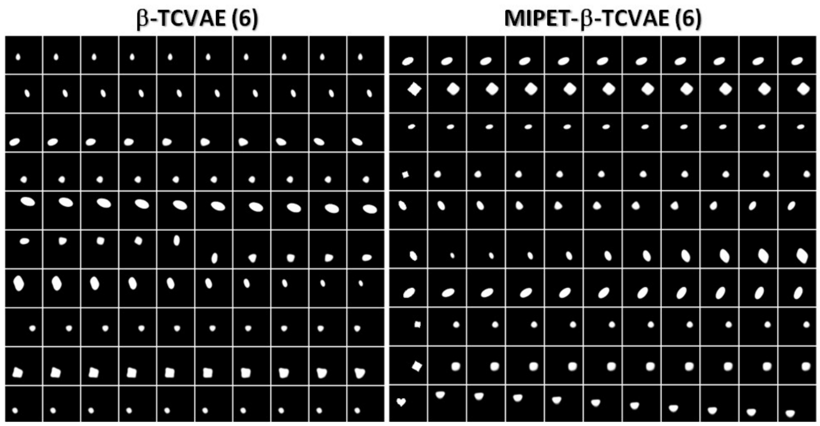
6.2 Qualitative Analysis

We randomly sample an image for each dimension of the latent vector space and creates 10 variants of its generated latent vector by selecting values from {-2, 2} with 10 intervals for the dimension, then generate their corresponding output images. For the generation, we select β-TCVAE (6), which shows the best FVM scores in dSprites dataset. Thereafter, we evaluate the semantic roles of each dimension before and after applying MIPE-transformation function.

In Figure 4, β-TCVAE struggles with y-position and rotation, as shown on the 6^{th} row, and with scale and shape represented on the 7^{th} row. On the contrary, MIPET-β-TCVAE separates y-position and rotation factor (10^{th}, and 7^{th} rows), also the activated dimensions of MIPET-β-TCVAE are not overlapped with each factor. Applied our method on β-TCVAE shows better disentangled representation on dSprites dataset. These results also show that our proposed method improves disentangled representation learning. As shown in the Figure 5, β-VAE struggles with rotation and scale factors in 4^{th}dimension. Also, it struggles with x-position and scale factors in 8^{th} dimension, and x-position and rotation factors in 9^{th} dimension. However, MIPET-β-VAE only struggles with rotation and shape factors in 5^{th} dimension. As shown in the Figure 6, CLG-VAE struggles with rotation and shape factors in 2^{nd} dimension, and shape and scale factors in 7^{th} dimension. However, MIPET-CLG-VAE separates rotation and shape factors in 10^{th}, and 1^{st} dimensions respectively.

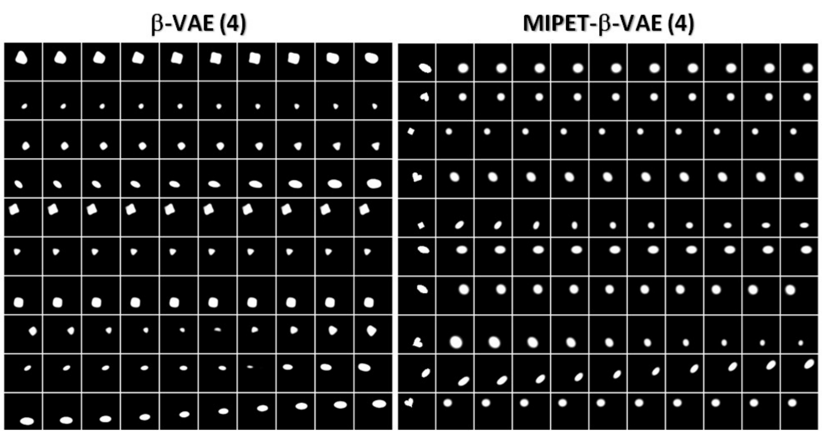
The qualitative analysis with 3D Shapes dataset, as shown in the Figure 7, β-VAE struggles with all factors, and only the object color factor is divided in 6^{th} dimension. However, this factor is still activated with scale factor in 3^{rd} dimension. Although MIPET-β-VAE struggles with reconstruction, it is less struggle with than β-VAE. As shown in the Figure 8, CLG-VAE struggles with shape and wall color factors in 4^{th} dimension, and shape and object color factors in 7^{th} dimension. In particular, it struggles with tree factors in 9^{th} dimension. On the other hand, MIPET-CLG-VAE separates shape, wall, and object color factors.

The qualitative analysis with 3D Cars dataset, as show in Figure 9, the left side is the β-TCVAE result, and it struggles with body, and azimuth factors shown in the 7^{th} row. However, MIPET-β-TCVAE separates azimuth (6^{th} row) and body (1^{st} row). In particular, MIPET-β-TCVAE learns *color* factor (3^{rd} row) which does not exist on β-TCVAE.



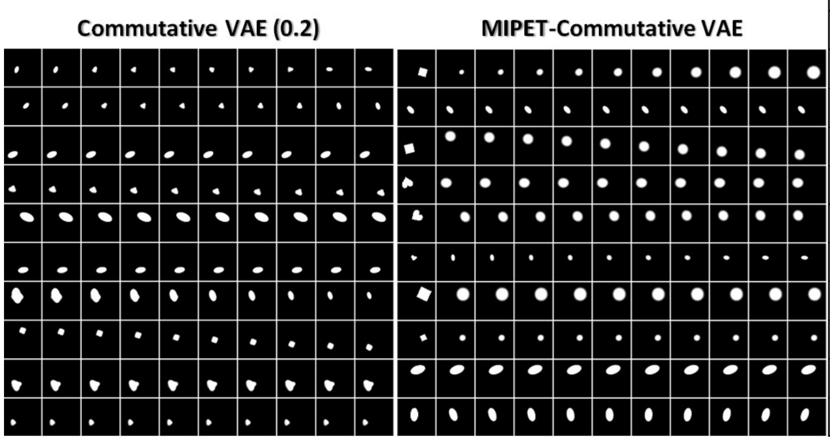
Factors						
X-pos	Y-pos	Rotation	Shape	Scale	# of factor	
					-	-
					-	-
		√			1	-
√					-	1
	√		√		1	1
	√	√		√	2	1
		√	√	√	2	1
√					1	-
			√		1	-
	√				-	1

Fig. 4: Qualitative results on dSprites. The left-side grids are input images and their variants by changing activations of each dimension of latent vectors. The first row shows input images. The right-side table shows matching pre-defined factors of the dataset (red: MIPET, blue: no MIPET).



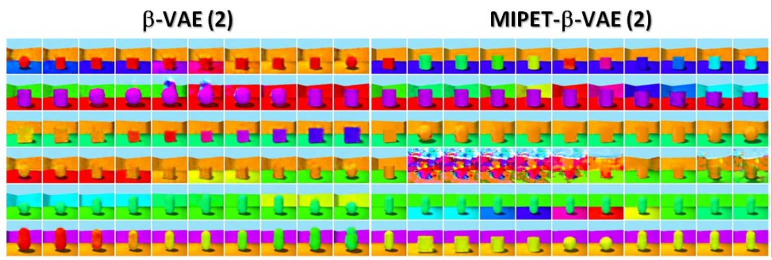
Factors						
X-pos	Y-pos	Rotation	Shape	Scale	# of factor	
			√		1	-
					-	-
		√			1	-
		√		√	2	-
		√	√		-	2
					-	-
√					-	1
√				√√	2	1
√	√	√			2	1
	√				1	-

Fig. 5: Qualitative analysis result of β-VAE and MIPET-β-VAE.



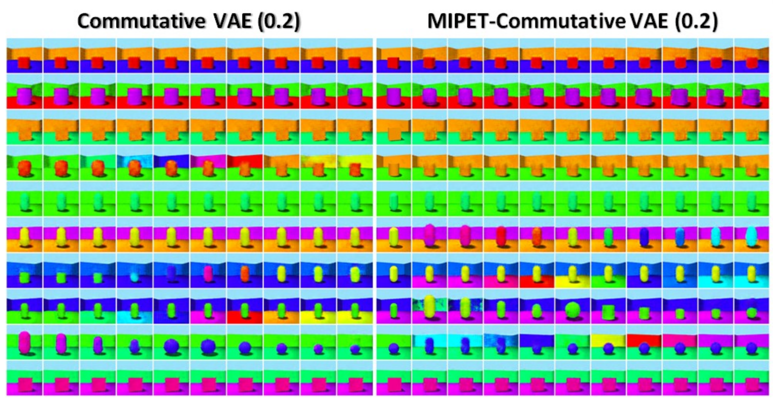
Factors						
X-pos	Y-pos	Rotation	Shape	Scale	# of factor	
		√	√		1	1
		√	√		2	-
	√				-	1
√					1	-
√					-	1
		√	√		-	2
			√	√	2	-
	√				1	-
					-	-
		√			-	1

Fig. 6: Qualitative analysis result of CLG-VAE (0.2) and MIPET-CLG-VAE (0.2) with dSprites.



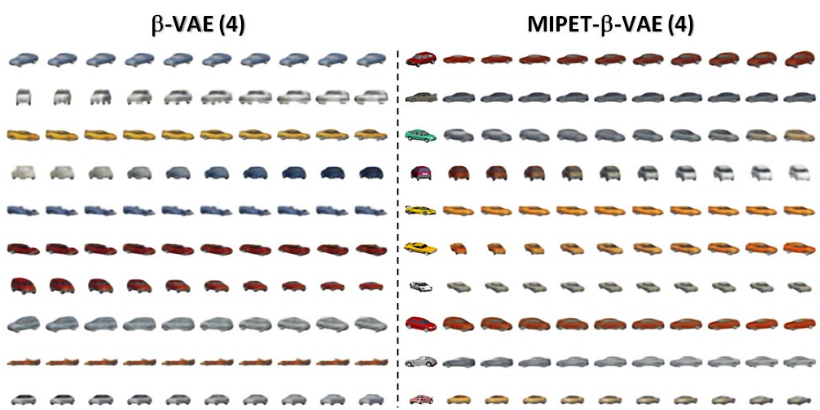
Factors							
Shape	Orien	Scale	Wall	Floor	Object	# of factor	
√√				√	√	2	2
√√			√√			2	2
√		√			√	2	1
√√	√			√		2	2
√	√		√	√		3	1
√√		√			√	1	2

Fig. 7: The Shape is object shape, Orien is an orientation of object, Scale is a scale factor of object, Wall is wall color factor, Floor is floor color, and Object is object color factors. It represents the β-VAE ($\beta = 2$) results.



Factors							
Shape	Orien	Scale	Wall	Floor	Object	# of factor	
	√					-	1
	√					1	-
						-	-
√			√			2	-
						-	-
					√	-	1
√				√	√	2	1
√		√		√		1	2
√		√	√		√	3	1
						-	-

Fig. 8: Qualitative analysis result of CLG-VAE (0.2) and MIPET-CLG-VAE (0.2) with 3D Shapes.



Factors						
Azimuth	Body	Whiten	Yaw	Color	# of factor	
	√				-	1
√					1	-
	√			√	1	1
		√√			1	1
					-	-
√√					1	1
√	√				2	-
			√√		1	1
		√			-	1
		√√			1	1

Fig. 9: Qualitative analysis result of β-VAE (4.0) with 3D Cars.

7 Conclusion

In this paper, we address the problem of injecting inductive bias for learning unsupervised disentangled representations. To build the bias in VAE frameworks, we propose MIPE-transformation composed of 1) IPE-transformation for the benefits of invertibility and partial-equivariant for disentanglement, 2) a training loss and module to adapt unrestricted prior and posterior to an approximated exponential family, and 3) integration of multiple units of IPE-transformation function and EF-conversion for more expressive bias. The method is easily equipped on state-of-the-art VAEs for disentanglement learning and shows significant improvement on dSprites, 3D Shapes, and 3D Cars datasets. We expect that our method can be applied to more VAEs, and extended to downstream applications. Our work is limited to holding partial equivariance of I2L transformation, so more direct methods to induce it can be integrated in the future.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (2022R1A2C2012054).

References

[1] S. Singla, B. Nushi, S. Shah, E. Kamar, and E. Horvitz, "Understanding failures of deep networks via robust feature extraction," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021*. Computer Vision Foundation / IEEE, 2021.

[2] J. Tenenbaum, "Building machines that learn and think like people," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, ser. AAMAS '18. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2018, p. 5.

[3] Y.-J. Li, C.-S. Lin, Y.-B. Lin, and Y.-C. F. Wang, "Cross-dataset person re-identification via unsupervised pose disentanglement and adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[4] Y. Zou, X. Yang, Z. Yu, B. Vijayakumar, and J. Kautz, "Joint disentangling and adaptation for cross-domain person re-identification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[5] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013.

[6] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *ICLR*, 2017.

[7] R. T. Q. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud, "Isolating sources of disentanglement in variational autoencoders," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018.

[8] H. Kim and A. Mnih, "Disentangling by factorising," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 2649–2658.

[9] Y. Jeong and H. O. Song, "Learning discrete and continuous factors of data via alternating disentanglement," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 3091–3099.

[10] Z. Li, J. V. Murkute, P. K. Gyawali, and L. Wang, "Progressive learning and disentanglement of hierarchical representations," in *International Conference on Learning Representations*, 2020.

[11] N. Dilokthanakul, P. A. M. Mediano, M. Garnelo, M. C. H. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan, "Deep unsupervised clustering with gaussian mixture variational autoencoders." *CoRR*, vol. abs/1611.02648, 2016.

[12] T. R. Davidson, L. Falorsi, N. De Cao, T. Kipf, and J. M. Tomczak, "Hyperspherical variational auto-encoders," *34th Conference on Uncertainty in Artificial Intelligence (UAI-18)*, 2018.

[13] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 4114–4124.

[14] N. Miao, E. Mathieu, S. N, Y. W. Teh, and T. Rainforth, "On incorporating inductive biases into VAEs," in *International Conference on Learning Representations*, 2022.

[15] X. Zhu, C. Xu, and D. Tao, "Commutative lie group vae for disentanglement learning," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 12 924–12 934.

[16] T. Yang, X. Ren, Y. Wang, W. Zeng, and N. Zheng, "Towards building a group-based unsupervised representation disentanglement framework," in *International Conference on Learning Representations*, 2022.

[17] I. Higgins, D. Amos, D. Pfau, S. Racanière, L. Matthey, D. J. Rezende, and A. Lerchner, "Towards a definition of disentangled representations," *CoRR*, vol. abs/1812.02230, 2018.

[18] D. W. Romero and S. Lohit, "Learning partial equivariances from data," in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022.

[19] Y. Yang, Y. Chen, and S. Soatto, "Learning to manipulate individual objects in an image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[20] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[21] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016.

[22] I. Jeon, W. Lee, M. Pyeon, and G. Kim, "Ib-gan: Disengangled representation learning with information bottleneck generative adversarial networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 9, 2021, pp. 7926–7934.

[23] C. Eastwood and C. K. I. Williams, "A framework for the quantitative evaluation of disentangled representations," in *International Conference on Learning Representations*, 2018.

[24] N. Watters, L. Matthey, C. P. Burgess, and A. Lerchner, "Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes," *CoRR*, vol. abs/1901.07017, 2019.

[25] T. A. Keller and M. Welling, "Topographic vaes learn equivariant capsules," *CoRR*, vol. abs/2109.01394, 2021.

[26] Y. Li and S. Mandt, "Disentangled sequential autoencoder," 2018.

[27] J. Bai, W. Wang, and C. P. Gomes, "Contrastively disentangled sequential variational autoencoder," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021.

[28] N. Dilokthanakul, P. A. M. Mediano, M. Garnelo, M. C. H. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan, "Deep unsupervised clustering with gaussian mixture variational autoencoders," *CoRR*, vol. abs/1611.02648, 2016.

[29] C. Xiao and L. Liu, "Generative flows with matrix exponential," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 10 452–10 461.

[30] E. Hoogeboom, V. Garcia Satorras, J. Tomczak, and M. Welling, "The convolution exponential and generalized sylvester flows," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 18 249–18 260.

[31] B. Charpentier, D. Zügner, and S. Günnemann, "Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts," in *NeurIPS*, 2020.

[32] B. Charpentier, O. Borchert, D. Zügner, S. Geisler, and S. Günnemann, "Natural Posterior Network: Deep Bayesian Predictive Uncertainty for Exponential Family Distributions," in *International Conference on Learning Representations*, 2022.

[33] R. Ranganath, L. Tang, L. Charlin, and D. M. Blei, "Deep exponential families," 2014.

[34] I. Higgins, S. Racanière, and D. Rezende, "Symmetry-based representations for artificial and biological general intelligence," 2022.

[35] W. Peebles, J. Peebles, J.-Y. Zhu, A. A. Efros, and A. Torralba, "The hessian penalty: A weak prior for unsupervised disentanglement," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

[36] I. Higgins, D. Amos, D. Pfau, S. Racanière, L. Matthey, D. J. Rezende, and A. Lerchner, "Towards a definition of disentangled representations," *CoRR*, vol. abs/1812.02230, 2018.

[37] B. Charpentier, O. Borchert, D. Zügner, S. Geisler, and S. Günnemann, "Natural posterior network: Deep bayesian predictive uncertainty for exponential family distributions," in *International Conference on Learning Representations*, 2022.

[38] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[39] P. Bader, S. Blanes, and F. Casas, "Computing the matrix exponential with an optimized taylor polynomial approximation," *Mathematics*, vol. 7, no. 12, 2019.

[40] L. Matthey, I. Higgins, D. Hassabis, and A. Lerchner, "dsprites: Disentanglement testing sprites dataset," https://github.com/deepmind/dsprites-dataset/, 2017.

[41] C. Burgess and H. Kim, "3d shapes dataset," https://github.com/deepmind/3dshapes-dataset/, 2018.

[42] S. E. Reed, Y. Zhang, Y. Zhang, and H. Lee, "Deep visual analogy-making," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015.

[43] X. Ren, T. Yang, Y. Wang, and W. Zeng, "Learning disentangled representation by exploiting pretrained generative models: A contrastive learning view," in *ICLR*, 2022.

[44] A. Kumar, P. Sattigeri, and A. Balakrishnan, "VARIATIONAL INFERENCE OF DISENTANGLED LATENT CONCEPTS FROM UNLABELED OBSERVATIONS," in *International Conference on Learning Representations*, 2018.

[45] N. J. Michlo, "Disent - a modular disentangled representation learning framework for pytorch," Github, 2021.

Hee-Jun Jung received the B.S. from the Kyung Hee University (KHU), South Korea, in 2020. He is currently in Ph.D. degree with the Gwangju Institute of Science and Technology (GIST) in South Korea. His research interests include variational auto-encoder, disentanglement learning, natural language processing, and knowledge distillation.

Jaehyoung Jeong received the B.S. from the Chung-Ang University, South Korea, in 2022. He is currently in Ph.D. degree with the Gwangju Institute of Science and Technology (GIST) in South Korea. His research interests include variational autoencoder, disentanglement learning, generative model, and geometric machine learning.

Kangil Kim received Ph.D. degree from Seoul National University, Seoul, Korea, in 2012. He worked as a senior researcher in ETRI, Daejon from 2013 to 2016, as an assistant professor at Konkuk university from 2016 to 2019. He is currently an assistant professor in Gwangju Institute of Science and Technology. His research interests include artificial intelligence, evolutionary computation, machine learning, and natural language processing.