# MUTIPLE INVERTIBLE AND EQUIVARIANT TRANSFORMATION FOR DISENTANGLEMENT IN VAES

**Anonymous authors**
Paper under double-blind review

## 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 Equivariant transformation* (MIE-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 Equivariant transformation* (IE-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 IE-transformation and EF-conversion, and their training. In experiments on 3D Cars, 3D Shapes, and dSprites datasets, MIE-transformation improves the disentanglement performance of state-of-the-art VAEs.

## 1 INTRODUCTION

Disentanglement learning to learn more interpretable representations is broadly useful in artificial intelligence fields such as classification (Singla et al., 2021), zero-shot learning (Tenenbaum, 2018), and domain adaptation (Li et al., 2019; Zou et al., 2020). 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) (Kingma & Welling, 2013). Based on VAE, unsupervised disentangled representation learning has been elaborated (Higgins et al., 2017; Chen et al., 2018; Kim & Mnih, 2018; Jeong & Song, 2019; Li et al., 2020) through the factorizable variations and control of uncorrelatedness of each dimension of representations. Moreover, VAE models to handle the shape of prior as a Gaussian mixture (Dilokthanakul et al., 2016) or von Mises-Fisher (Davidson et al., 2018) 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 (Locatello et al., 2019).

Recently, such inductive bias has been introduced in various perspectives on transformation of latent vector space. Intel-VAE (Miao et al., 2021) 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 of disentanglement (Zhu et al., 2021; Yang et al., 2021), whose definition follows Higgins et al. (2018a), 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.

Inspired by the above works, we propose a *Multiple Invertible and Equivariant transformation* (MIE-transformation) method[1], which is simply insertable to VAEs. The method adopts the matrix exponential to hold the invertible property of latent-to-latent (L2L) vector transformation. Then, to preserve at least some potential equivariance between input-to-latent (I2L) vector transformation, we constrain the L2L transformation to a symmetric matrix exponential, called *invertible and*

[1] available on Github, which will be released after publication.

*equivariant transformation* (IE-transformation). The IE-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 IE-transformation work as $\beta$ parameters (Higgins et al., 2017) controlled for enhancing disentanglement learning. Also, we propose the *sparse log-normalizer* to induce an implicit semantic mask in the latent vector space, different to Yang et al. (2020). In experiments with quantitative and qualitative analysis, MIE-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 MIE-transformation architecture to integrate multiple IE-transformation and EF-conversion, which is widely applicable to state-of-the-art VAEs.
4. We empirically analyze the properties of MIE-transformation and validate its effectiveness in disentanglement learning on benchmarks.

## 2 Related Work

Recently, various works have focused on the unsupervised disentanglement learning. One of the branches is InfoGAN (Chen et al., 2016) based works such as IB-GAN (Jeon et al., 2021) and OOGAN (Liu et al., 2020) showed the improvements, but these works regularize informativeness introduced in Lin et al. (2020) to elaborate regularizing MI. The other branch is based on the Variational AutoEncoder (VAE). $\beta$-VAE (Higgins et al., 2017) penalizes Kullback-Leibler divergence (KL divergence) with weighted hyper-parameters. Factor VAE (Kim & Mnih, 2018) and $\beta$-TCVAE (Chen et al., 2018) are trained with total correlation (TC) to make independent dimensions on a latent vector with discriminator and divided KL divergence term. Differently, we consider the impact based on group theory based on Higgins et al. (2018a).

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) (Zhu et al., 2021) 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 (Yang et al., 2021) extends Spatial Broadcast Decoder (Watters et al., 2019) with the group theory, and it also proves the necessity of the proposed abel loss and order loss with cyclic groups and n-th root unity group to improve disentangled representation. Topographic VAE (Keller & Welling, 2021) 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.

Other works elaborate uncertainty of exponential family distribution with Bayesian update (Charpentier et al., 2020; 2022b). Other approaches in VAEs, some works have implemented extension prior such as a transformed Gaussian distribution, Gaussian mixture distribution (Kalatzis et al., 2018) or von Mises-Fisher distribution (Davidson et al., 2018). InteL-VAE (Miao et al., 2021) 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 Xiao & Liu (2020), utilized matrix exponential on Neural networks is invertible, but it only provides mathematical foundations of the transformation. Matrix exponential flows are proposed in Hoogeboom et al. (2020) for equivariant simultaneously. In our work, we show how to use it for disentanglement learning.

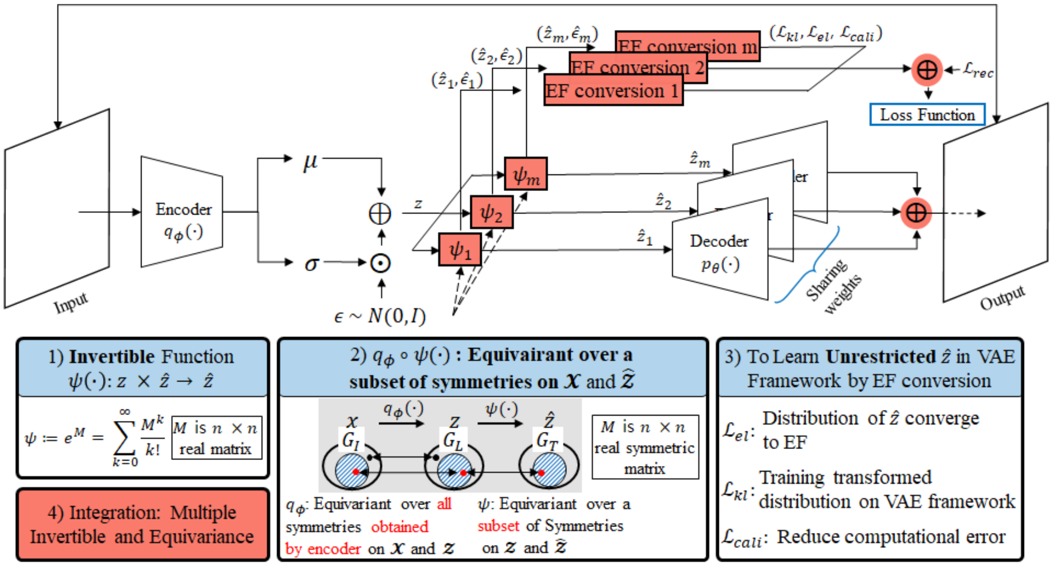

$(\hat{z}_m, \hat{\epsilon}_m)$
EF conversion m
$(\mathcal{L}_{kl}, \mathcal{L}_{el}, \mathcal{L}_{cali})$
$(\hat{z}_2, \hat{\epsilon}_2)$
EF conversion 2
$(\hat{z}_1, \hat{\epsilon}_1)$
EF conversion 1
$\mathcal{L}_{rec}$
Loss Function
$\mu$
Encoder
$q_\phi(\cdot)$
$z$
$\psi_m$
$\psi_2$
$\psi_1$
$\hat{z}_m$
$\hat{z}_2$
$\hat{z}_1$
$\sigma$
$\epsilon \sim N(0, I)$
Decoder
$p_\theta(\cdot)$
Sharing weights
Input
Output
1) Invertible Function
$\psi(\cdot): z \times \hat{z} \rightarrow \hat{z}$
$\psi := e^M = \sum_{k=0}^{\infty} \frac{M^k}{k!}$
$M$ is $n \times n$ real matrix
4) Integration: Multiple Invertible and Equivariance
2) $q_\phi \circ \psi(\cdot)$ : Equivairant over a subset of symmetries on $\mathcal{X}$ and $\hat{\mathcal{Z}}$
$x$
$q_\phi(\cdot)$
$z$
$\psi(\cdot)$
$\hat{z}$
$G_I$
$G_L$
$G_T$
$M$ is $n \times n$ real symmetric matrix
$q_\phi$: Equivariant over all symmetries obtained by encoder on $\mathcal{X}$ and $\mathcal{Z}$
$\psi$: Equivariant over a subset of Symmetries on $\mathcal{Z}$ and $\hat{\mathcal{Z}}$
3) To Learn Unrestricted $\hat{z}$ in VAE Framework by EF conversion
$\mathcal{L}_{el}$: Distribution of $\hat{z}$ converge to EF
$\mathcal{L}_{kl}$: Training transformed distribution on VAE framework
$\mathcal{L}_{cali}$: Reduce computational error

Figure 1: The overall architecture of our proposed *MIET*-VAE. The invertible and equivariant function $\psi(\cdot)$ consists of a symmetric matrix exponential to be 1) invertible and 2) partially equivariant. Then 3) EF conversion module converges the distribution of unrestricted $\hat{z}$ to the power density function of 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 details of the gray box are in Fig. 2a.

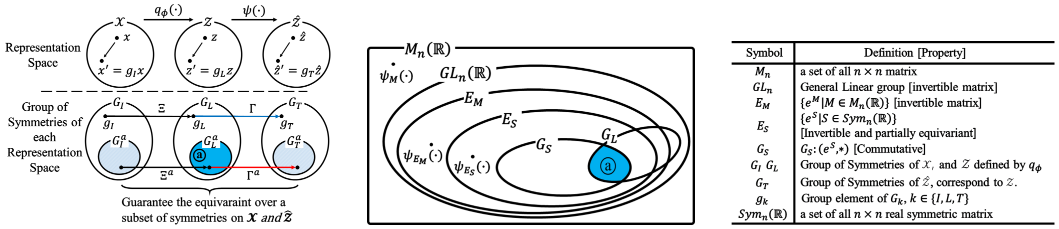
## 3 Method

The overview of a VAE equipped with MIE-transformation is shown in Fig. 1. The MIE-transformation has three main components: 1) *IE-Transformation Unit* to transform latent vectors with invertible and 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 IE-transformation and EF-conversion.

### 3.1 Invertible and Equivariant Transformation

**Invertible Property by Using Matrix Exponential** To guarantee the invertible property of IE-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 $M_n(\mathbb{R})$ (Xiao & Liu, 2020). The operator $*$ is matrix multiplication, and $\mathbf{e}^{\boldsymbol{M}} = \sum_k^\infty \frac{\boldsymbol{M}^k}{k!}$. 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 invertible transformation function (Miao et al., 2021). Our motivation is to use the benefits of injecting explicit inductive bias for disentanglement Locatello et al. (2019); Miao et al. (2021).

**Why Should L2L Transformation Be Equivariant?** Let's consider equivariance between the input space and the final latent vector space directly used for a decoder in the VAE frameworks. We assume that previous works are equivariant over a subset of symmetries on the input and the latent vector space because these methods have improved disentanglement learning. However, if we apply the unrestricted L2L transformation to the VAE, then there is no guarantee to be equivariant between the input and final space. This problem is more precisely shown in Fig. 2a, which illustrates equivariance 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 equivairant.



(a) Space and symmetries.

(b) Venn diagram of each group.

| Symbol | Definition [Property] |
|---|---|
| $M_n$ | a set of all $n \times n$ matrix |
| $GL_n$ | General Linear group [invertible matrix] |
| $E_M$ | $\{e^M \vert M \in M_n(\mathbb{R})\}$ [invertible matrix] |
| $E_S$ | $\{e^S \vert S \in Sym_n(\mathbb{R})\}$<br>[Invertible and partially equivariant] |
| $G_S$ | $G_S$: $(e^S, *)$ [Commutative] |
| $G_I$ $G_L$ | Group of Symmetries of $\mathcal{X}$, and $\mathcal{Z}$ defined by $q_\phi$ |
| $G_T$ | Group of Symmetries of $\hat{\mathcal{Z}}$, correspond to $\mathcal{Z}$. |
| $g_k$ | Group element of $G_k$, $k \in \{I, L, T\}$ |
| $Sym_n(\mathbb{R})$ | a set of all $n \times n$ real symmetric matrix |

(c) Defitnition of each symbol.

Figure 2: $G_I$, and $G_L$ are obtained through $q_\phi$. Fig. 2a shows the relation between each space and symmetries. If $q_\phi$, and $\psi(\cdot)$ is equivariant function over all $G_I$, $G_L$, and $G_T$, then there exist $\Xi$, and $\Gamma$, respectively, where $\Xi : G_I \to G_L$, and $\Gamma : G_L \to G_T$, and $\Xi \circ \Gamma : G_I \to G_T$. However, unrestricted $\psi(\cdot)$ has no guarantee to be equivariant. The red arrows represent our method.

**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 $\psi(\cdot)$ in the group, 2) $\psi(\cdot)$ being equivariant over subset of symmetries between the input transformed latent vector space, and 3) the constraint increases the probability of $\psi(\cdot)$ to be in the group.

We distinguish $M_n(\mathbb{R})$, $E_M$, $E_S$, and $G_S$ as shown in Fig 2b, and 2c. The intersection ⓐ : $G_S \cap G_L$ is shown in Fig. 2a, and 2b. We particularly call the transformations as *symmetries* (Higgins et al., 2022) to distinguish them from IE- 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 Fig. 2b.

**Proposition 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 $\psi_{G_S}$, $g_s$, and $G_S$, respectively. ■

**Proposition 2.** *If $q_\phi$ 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 symmetryies in $G_I$ corresponding to ⓐ and $G_T$ corresponding to ⓐ by the equivariance of $q_\phi$.*

*Proof.* The function $\psi_{G_S}(\cdot)$ is an equivariant function over group elements in ⓐ by Proposition 1. Then, the composite function, $\psi_{G_S}(\cdot)$ and $q_\phi$, is an equivariant function of $G_I$ corresponding to ⓐ and $G_T$ corresponding to ⓐ (see Appendix C.2). ■

In other words, $\psi_{G_S}(\cdot)$ guarantees to preserve the equivariance of I2L-transformation to certain symmetries in ⓐ after IE-transformation.

Let $P(B)$ be the probability of $\psi(\cdot) \in B$ for a subset $B \subset M_n(\mathbb{R})$ after VAE training. Then,

**Proposition 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 (see Appendix C.1). Therefore, the probability $Pr(\psi_{E_M}(\cdot) \in G_S) = \frac{P(G_S)}{P(E_M)}$ is greater than $Pr(\psi_{E_S}(\cdot) \in G_S) = \frac{P(G_S)}{P(E_S)}$. 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, $\psi_{E_S}$ clearly increases the probability of preserving a certain type of equivariance compared to unrestricted $\psi$ functions.

| dSprites | Disentanglement Metric | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | FVM ↑ | | MIG ↑ | | SAP ↑ | | DCI ↑ | |
| | original | MIET | original | MIET | original | MIET | original | MIET |
| $\beta$-VAE | 69.15(±5.88) | **74.19**(±5.62) | 9.49(±8.30) | **19.72**(±11.37) | 2.43(±2.07) | **5.08**(±2.90) | 18.57(±12.41) | **28.81**(±10.19) |
| $\beta$-TCVAE | 78.50(±7.93) | **79.87**(±5.80) | 26.00(±9.06) | **35.04**(±4.07) | 7.31(±0.61) | **7.70**(±1.63) | 41.80(±8.55) | **47.83**(±5.01) |
| CLG-VAE | 79.06(±6.83) | **81.80**(±3.17) | 23.40(±7.89) | **36.34**(±5.55) | 7.37(±0.96) | **8.03**(±0.83) | 37.68(±7.83) | **44.73**(±5.11) |

| 3D Shapes | Disentanglement Metric | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | FVM ↑ | | MIG ↑ | | SAP ↑ | | DCI ↑ | |
| | original | MIET | original | MIET | original | MIET | original | MIET |
| $\beta$-VAE | 71.76(±12.26) | **75.19**(±8.16) | 37.33(±22.34) | **47.37**(±10.13) | 7.48(±4.12) | **9.20**(±2.44) | 52.07(±17.92) | **54.95**(±8.99) |
| $\beta$-TCVAE | 76.62(±10.23) | **80.59**(±8.57) | 52.93(±20.5) | **54.49**(±9.44) | 10.64(±5.93) | **11.58**(±3.32) | 65.32(±11.37) | **66.22**(±7.32) |
| CLG-VAE | 77.04(±8.22) | **80.17**(±8.43) | 49.74(±8.18) | **53.87**(±7.41) | 9.20(±2.44) | **12.83**(±3.01) | 57.70(±8.60) | **60.74**(±7.77) |

| 3D Cars | Disentanglement Metric | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | FVM ↑ | | MIG ↑ | | SAP ↑ | | DCI ↑ | |
| | original | MIET | original | MIET | original | MIET | original | MIET |
| $\beta$-VAE | **89.48**(±5.22) | 88.95(±5.94) | 6.90(±2.70) | **7.27**(±1.99) | 1.30(±0.48) | **1.88**(±1.12) | **19.85**(±4.87) | 18.90(±4.49) |
| $\beta$-TCVAE | 95.84(±3.40) | **96.43**(±2.42) | **11.87**(±2.90) | 10.80(±1.22) | 1.55(±0.38) | **1.88**(±1.12) | **27.91**(±4.31) | 26.08(±2.47) |
| CLG-VAE | 86.11(±7.12) | **91.06**(±5.09) | 6.19(±2.42) | **8.51**(±2.11) | **2.06**(±0.60) | 1.99(±0.93) | 16.91(±4.01) | **18.31**(±2.83) |

Table 1: Performance (mean ± std) of four metrics on dSprites, 3D Shapes, and 3D Cars. The $\alpha = 1$ and $\gamma = 1$ of $\beta$-TCVAE as Chen et al. (2018).

| *p*-value | VAEs | | | | CLG-VAE | | | | $\beta$-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 |

Table 2: *p*-value of t-test for original vs MIET results of Table 1, which are averaged over models (bold: positive and significant, italic: positive but insignificant, normal: lower performance).

In practice, however, there are uncertain and undefinable conditions to derive the total probability of preserving all existing equivariance. For example, probability distribution $P(\cdot)$ varies by training settings, so Proposition 3 holds with only uniform or equal distributions determined by training settings for $\psi_{E_M}(\cdot)$ and $\psi_{E_S}(\cdot)$. Additionally, the area of $G_L \backslash G_S$ and its probability are uncertain and depend on I2L transformation functions such as encoders of VAEs. We empirically validate the impact of equivariance with the uncertain $P(\cdot)$ to disentanglement in Section 5.1.

**Independence of Latent Variables** In addition to invertible and equivariant properties, our IE-transformation also guarantees zero Hessian matrix, which enhances disentanglement without any additional loss of Peebles et al. (2020). Hessian matrix of the transformation $\nabla^2_{\boldsymbol{z}}\psi(\boldsymbol{z}) = \nabla_{\boldsymbol{z}}(\nabla_{\boldsymbol{z}}\mathbf{e}^{\boldsymbol{M}}\boldsymbol{z}) = 0$ because of the irrelevance of $\boldsymbol{M}$ to $\boldsymbol{z}$. By this property, $\psi(\cdot)$ leads that independently factorizes each dimension (Peebles et al., 2020), 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 $j \neq i$ (Yang et al., 2021; Higgins et al., 2018b). This correspondence of decomposition is expected to transfer the independence between dimensions of $\boldsymbol{z}$ to the space of $\hat{\boldsymbol{z}}$ (Higgins et al., 2018a).

### 3.2 Exponential Family Conversion

The generated latent variables by IE-transformation may not follow a Gaussian distribution generally used for the training of VAEs in a non-parametric perspective. 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 IE-transformation function and EF-conversion, and more details of each procedure are in Appendix B.

**Posterior Approximation As an Exponential Family** The procedure represents a posterior distribution in the exponential family by adopting the following form:

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

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 $\xi$ (Bishop,

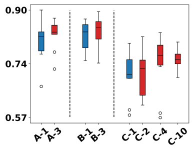
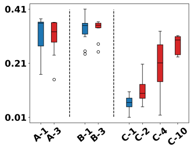
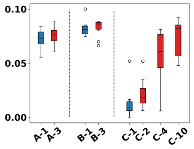
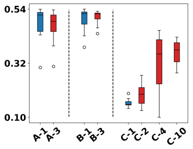
| dataset | Metrics | | | |
|---|---|---|---|---|
| | FVM | IMG | SAP | DCI |
| dSprites |  |  |  |  |

Table 3: Impact of the number of MIE-transformation function on the $\beta$-TCVAE and $\beta$-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 IE-transformation cases, respectively. (A-$n$: MIET-$\beta$-TCVAE (4), B-$n$: MIET-$\beta$-TCVAE (6), C-$n$: MIET-$\beta$-VAE, $n$: the number of MIE-transformation)

2006). 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 Charpentier et al. (2022a). 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 IE-transformation, we can apply the form by using the $\hat{\boldsymbol{z}}_m$, $\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m}$, and $\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}_m}$ for $\mathbf{X}$, $\boldsymbol{\theta}$, and $\xi$, respectively. More details of the background, conjugate prior, posterior, and assumptions are in Appendix B.3.

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

$$\text{maximize } \log p(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m} | \hat{\boldsymbol{z}}_m, \boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}_m}, \boldsymbol{\nu}_m) \quad \text{s.t. } D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m}) || f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}_m})) \geq 0 \tag{2}$$

$$\Rightarrow \mathcal{L}_s(\hat{\boldsymbol{z}}_m, \hat{\boldsymbol{\epsilon}}_m) = \log p(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m} | \hat{\boldsymbol{z}}_m, \boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}_m}, \boldsymbol{\nu}_m) + \boldsymbol{\lambda}_m D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m}) || f_{\mathbf{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}_m})) \tag{3}$$

$$\Rightarrow \mathcal{L}_{el} \coloneqq ||\nabla_{\hat{\boldsymbol{z}}_m, \hat{\boldsymbol{\epsilon}}_m, \boldsymbol{\lambda}_m} \mathcal{L}_s||_2^2 = 0. \tag{4}$$

This provides EF similarity loss in Eq. 9. The notation $\boldsymbol{\theta}_k$ is a generated natural parameter by a given $k \in \{\hat{\boldsymbol{z}}, \hat{\boldsymbol{\theta}}\}$, 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 to guarantee KL divergence of the transformed distribution always being positive. More details of Eq. 4 are in Appendix B.3.

**KL Divergence for Evidence of Lower Bound** The KL divergence of Gaussian distribution (Kingma & Welling, 2013) 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. 1 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:

$$\mathcal{L}_{kl} \coloneqq D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m}) || f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}_m})) = A(\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}) - A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) + \boldsymbol{\theta}_{\hat{\boldsymbol{z}}}^{\intercal} \nabla_{\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}} A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) - \boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}^{\intercal} \nabla_{\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}} A(\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}). \tag{5}$$

More details of Eq. 5 are in Appendix B.2.

**KL divergence calibration loss** To reduce the error between the approximation and true matrix for the matrix exponential[2] (Bader et al., 2019), we add a loss to minimize the difference of their KL divergence measured by mean squared error (MSE) as:

$$\mathcal{L}_{cali} = \text{MSE}(D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) || p_\theta(\boldsymbol{z})), D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m}) || f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}_m}))), \tag{6}$$

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

**Sparse log-normalizer** We propose sparse log-normalizer 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:

$$\mathcal{M}_{ij} = \begin{cases} 1 \text{ if } |\mathcal{W}_{ij}| \geq \mu_{|\mathcal{W}_{ij}|} - \lambda \sigma_{|\mathcal{W}_{ij}|} \\ 0 \text{ otherwise} \end{cases}, \tag{7}$$

[2] https://pytorch.org/docs/1.10/generated/torch.matrix_exp.html#torch.matrix_exp

where $\mathcal{W}$ is the weight of log-normalizer, $\lambda$ is a hyper-parameter, $\mu_{|\mathcal{W}_{ij}|}$, and $\sigma_{|\mathcal{W}_{ij}|}$ are the mean, and standard deviation of weight respectively. Previous work (Yang et al., 2020) utilizes a semantic mask in input space directly, but we inject the semantic mask implicitly on the latent space.

### 3.3 Integration for Multiple IE-Transformation and EF-Conversion

We mathematically extend IE-transformation to MIE-transformation, which is the equivalent process of $\beta$-VAE to enhance disentanglement. The equivalence is proven in Appendix B.1. Each IE-transformation function operates independently, then $p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \cdots, \hat{\boldsymbol{z}}_k) = \prod_{i=1}^{k} p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_i)$. Therefore, the reconstruction error term ($\mathcal{L}_{rec}$) for given all $\hat{\mathbf{z}}_i$ in Eq. 9 is

$$\mathcal{L}_{rec} := \sum_{i=1}^{k} \left[ \int q_i(\hat{\boldsymbol{z}}_i|\boldsymbol{x}) \log p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_i) \mathrm{d}\hat{\boldsymbol{z}}_i \prod_{j=1, j\neq i}^{k} \int q_j(\hat{\boldsymbol{z}}_j|\boldsymbol{x}) \mathrm{d}\hat{\boldsymbol{z}}_j \right] = \sum_{i=1}^{k} E_{q_{\phi,\psi_i}(\boldsymbol{z}|\boldsymbol{x})} \log p_\theta(\boldsymbol{x}|\psi_i(\boldsymbol{z})), \tag{8}$$

where $\hat{\boldsymbol{z}}_i = \psi_i(\boldsymbol{z})$.However, according to the following Eq. 8, $k$ 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 of disentanglement from $k$ samples, as shown in Fig. 1. More derivation details of Eq. 8 are in Appendix B.1.

The VAEs equipped with MIE-transformation (MIET-VAEs) can be trained with the following loss:

$$\mathcal{L}(\phi, \theta, \psi_{i\in[1,k]}; \boldsymbol{x}) = \frac{1}{k}\mathcal{L}_{rec} - \mathcal{L}_{kl} - \mathcal{L}_{el} - \mathcal{L}_{cali}. \tag{9}$$

More derivation details of our proposed evidence of lower bound (ELBO) in Appendix B.

## 4 Experiment Settings

**Models** As baseline models, we select VAE, $\beta$-VAE, $\beta$-TCVAE, and CLG-VAE. These models are compared to their extension to adopt MIET, abbreviated by adding the MIET prefix. We apply the proposed method to $\beta$-TCVAE only with the EF similarity loss term because $\beta$-TCVAE penalizes the divided KL divergence terms. We set the same encoder and decoder architecture in each model to exclude the overlapped effects. More details of model architecture are in Table 8-9, and Appendix D.

**Datasets** We use following datasets: dSprites (Matthey et al., 2017), 3D Shapes (Burgess & Kim, 2018), and 3D cars (Reed et al., 2015). More information of datasets are in Appendix E.

**Training** We set 256 mini-batch size in the datasets (dSprites, 3D Shapes, and 3D Cars), Adam optimizer with learning rate $4 \times 10^{-4}$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$ 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 Kim & Mnih (2018); Ren et al. (2022). More details of the setting are in Appendix D.

**Evaluation** We evaluate four disentanglement metrics for a less biased understanding of the actual states of disentanglement. The used metrics include FactorVAE metric (FVM) (Kim & Mnih, 2018), MIG metric (Chen et al., 2018), SAP metric (Kumar et al., 2018), and DCI metric (Eastwood & Williams, 2018). More details of the evaluation metric are in Appendix F.

## 5 Results and Discussion

### 5.1 Quantitative Analysis

**Disentanglement Metrics** We set the number of IE-transformation functions to be equal to balancing hyper-parameter $\beta$ on $\beta$-VAE because of Eq. 9. The number of IE-transform functions of $\beta$-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 $\beta$-VAE. We average each

| 3D Cars | $\beta$-VAE | | | $\beta$-TCVAE | | |
|---|---|---|---|---|---|---|
| | MIET | MIET<br>(w/o E) | MIET<br>(w/o EF) | MIET | MIET<br>(w/o E) | MIET<br>(w/o EF) |
| FVM ↑ | **88.95**($\pm$5.94) | 82.09($\pm$11.33) | 45.23($\pm$6.39) | **96.43**($\pm$2.42) | 91.34($\pm$4.75) | 91.43($\pm$4.86) |
| MIG ↑ | **7.27**($\pm$1.99) | 6.77($\pm$2.41) | 0.04($\pm$0.02) | **10.80**($\pm$1.22) | 9.79($\pm$1.07) | 9.81($\pm$1.10) |
| SAP ↑ | **1.88**($\pm$1.12) | 1.76($\pm$1.06) | 0.18($\pm$0.12) | **1.88**($\pm$1.12) | 1.35($\pm$0.30) | 1.35($\pm$0.30) |
| DCI ↑ | **18.90**($\pm$4.49) | 17.21($\pm$5.57) | 1.67($\pm$1.26) | **26.08**($\pm$2.47) | 25.12($\pm$3.72) | 25.16($\pm$3.82) |

Table 4: 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 $\beta$-VAE and $\beta$-TCVAE, respectively.

| mask | $\beta$-VAE (1) | | | | CLG-VAE (0.5) | | | |
|---|---|---|---|---|---|---|---|---|
| ratio | FVM ↑ | MIG ↑ | SAP ↑ | DCI ↑ | FVM ↑ | MIG ↑ | SAP ↑ | DIC ↑ |
| 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) |
| $\infty$ | 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) |

Table 5: Impact of the mask (mean±std.) and its ratio $\lambda$ in Eq. 7 on 3D Cars. ($\infty$: 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.

model performance value with 40, 20, and 30 cases in VAEs, $\beta$-TCVAEs, and Commutative Lie Group VAE (CLG-VAEs) respectively.

As shown in Table 1, MIET-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 MIE-transformation functions on VAEs elaborate disentangled representation learning.

We additionally estimate the *p*-value of each metrics over models in Table 2. Previous work shows the average case of each models (Yang et al., 2021).

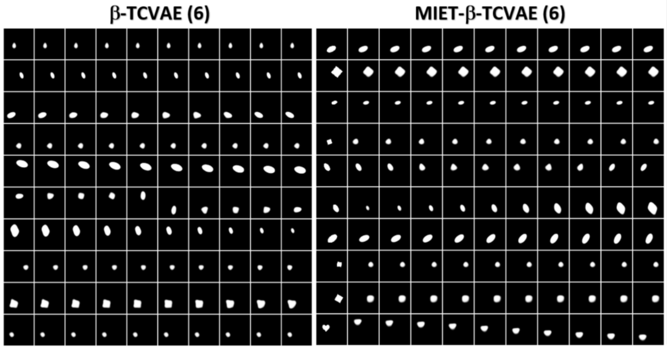
We divide each case into four categories: 1) Positive & Significant, 2) Positive & Insignificant, 3) Negative & Insignificant, and 4) Negative & Significant, where positive is when the mean value is higher than baseline and significant is statistically significant. We estimate the probability of each category: 1) 50%, 2) 36.11%, and 3) 13.89%. As shown in Table 2 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. $\beta$-TCVAEs is partially using our method (paragraph Models in Section 4), so it does not show the whole effect of MIET, but it improves model performance in many cases.

**Sensitivity to the Number of IE-transformation and EF-conversion** We investigate the impact of the MIE-transformation function. As presented in Table. 3, MIE-transformation is better than IE-transformation for disentanglement learning on each dataset. Indeed, MIET-$\beta$-VAEs results more clearly show the impact of the MIE-transfomation function. Our derivation in Section 3.3 and Appendix B clearly explains MIE-transformation impact. This result shows the impact of the multiple uses of IE-transformation and EF-conversion. More details are in Appendix H.

| dSprites | 3D Shapes | 3D Cars |
|---|---|---|
| 0.58 | 0.56 | 0.67 |

Table 6: The ratio of seeds to show better performance with symmetric matrix

**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 IE-transform and EF-conversion in the previous paragraph. We evaluate the impact of the other properties by setting MIE-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 $\beta$-VAE and $\beta$-TCVAE. As the results are shown in Table 4, most of the results show that MIET-VAEs performance is better than other cases. In particular, MIET (w/o EF) results are lower than MIET (w/o E) results and are clearly shown in all cases. More details are in Appendix I.



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

Figure 3: dSprites

Figure 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: MIET, blue: no MIET).

**Impact of Symmetric Matrix Exponential** We empirically show the benefit of using a symmetric matrix for $\psi$. Table 6 shows the ratio of runs with a symmetric matrix, which shows better performance than unrestricted matrices, to the total 240 (60 models × 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.

**Impact of Sparse Log-normalizer** We set masking hyper-parameter $\lambda$ from $\{0.0, 0.5, \cdots, 2.0, \infty\}$, and each model has different $\lambda$ for best case. In Table 5, VAE and CLG-VAE with masked log-normalizer show better and well-balanced results than the models without masking, which implies improvement of disentanglement.

### 5.2 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 $\beta$-TCVAE (6), which shows the best FVM scores in dSprites dataset. Thereafter, we evaluate the semantic roles of each dimension before and after applying MIE-transformation function.

In Fig. 3, $\beta$-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, MIET-$\beta$-TCVAE separates y-position and rotation factor ($10^{th}$, and $7^{th}$ rows), also the activated dimensions of MIET-$\beta$-TCVAE are not overlapped with each factor. Applied our method on $\beta$-TCVAE shows better disentangled representation on dSprites dataset. These results also show that our proposed method improves disentangled representation learning. More results are in Appendix K.

## 6 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 MIE-transformation composed of 1) IE-transformation for the benefits of invertibility and equivariance in 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 IE-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 potential equivariance of I2L transformation, so more direct methods to induce it can be integrated in the future.

## 7 Reproducibility Statement

In this section, we summarize detail contents for reproducing our theoretical and empirical results.

- Equations for MIET Method Implementation
    1. Objective function: Equation 9, Appendix B.1.
    2. EF family equation: Equation 1, Appendix B.3.
    3. EF similarity loss: Equation 2-4.
    4. KL Divergence of EF family: Equation 5, Appendix B.2.
    5. Reconstruction loss: Equation 8, Appendix 9.
- Motivation Proofs
    1. Proposition 1-3 : Appendix C.
- Experiment setting details
    1. Hyper-parameters: Appendix D.1.
    2. Training procedure and model configuration: Appendix D.2.
    3. Datasets: Appendix E.
    4. Quantitative analysis setting: Appendix F.
    5. Code description for running experiments: Supplementary material.

## References

Philipp Bader, Sergio Blanes, and Fernando Casas. Computing the matrix exponential with an optimized taylor polynomial approximation. *Mathematics*, 7(12), 2019. ISSN 2227-7390. doi: 10.3390/math7121174. URL https://www.mdpi.com/2227-7390/7/12/1174.

Junwen Bai, Weiran Wang, and Carla P Gomes. Contrastively disentangled sequential variational autoencoder. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=rWPxhfz2_S.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

Chris Burgess and Hyunjik Kim. 3d shapes dataset. https://github.com/deepmind/3dshapes-dataset/, 2018.

Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts. In *NeurIPS*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/0eac690d7059a8de4b48e90f14510391-Abstract.html.

Bertrand Charpentier, Oliver Borchert, Daniel Zügner, Simon Geisler, and Stephan Günnemann. Natural posterior network: Deep bayesian predictive uncertainty for exponential family distributions. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=tV3N0DWMxCg.

Bertrand Charpentier, Oliver Borchert, Daniel Zügner, Simon Geisler, and Stephan Günnemann. Natural Posterior Network: Deep Bayesian Predictive Uncertainty for Exponential Family Distributions. In *International Conference on Learning Representations*, 2022b.

Ricky T. Q. Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/1ee3dfcd8a0645a25a35977997223d22-Paper.pdf.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/7c9d0b1f96aebd7b5eca8c3edaa19ebb-Paper.pdf.

Tim R. Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M. Tomczak. Hyperspherical variational auto-encoders. *34th Conference on Uncertainty in Artificial Intelligence (UAI-18)*, 2018.

Nat Dilokthanakul, Pedro A. M. Mediano, Marta Garnelo, Matthew C. H. Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *CoRR*, abs/1611.02648, 2016. URL http://dblp.uni-trier.de/db/journals/corr/corr1611.html#DilokthanakulMG16.

Cian Eastwood and Christopher K. I. Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=By-7dz-AZ.

Robert M. Guralnick and Geoffrey R. Robinson. On the commuting probability in finite groups. *Journal of Algebra*, 300(2):509–528, 2006. ISSN 0021-8693. doi: https://doi.org/10.1016/j.jalgebra.2005.09.044. URL https://www.sciencedirect.com/science/article/pii/S0021869305007179. Special issue celebrating the 70th birthday of Bernd Fischer.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

Irina Higgins, David Amos, David Pfau, Sébastien Racanière, Loïc Matthey, Danilo J. Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *CoRR*, abs/1812.02230, 2018a. URL http://arxiv.org/abs/1812.02230.

Irina Higgins, David Amos, David Pfau, Sébastien Racanière, Loïc Matthey, Danilo J. Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *CoRR*, abs/1812.02230, 2018b. URL http://arxiv.org/abs/1812.02230.

Irina Higgins, Sébastien Racanière, and Danilo Rezende. Symmetry-based representations for artificial and biological general intelligence, 2022. URL https://arxiv.org/abs/2203.09250.

Emiel Hoogeboom, Victor Garcia Satorras, Jakub Tomczak, and Max Welling. The convolution exponential and generalized sylvester flows. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 18249–18260. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/d3f06eef2ffac7faadbe3055a70682ac-Paper.pdf.

Insu Jeon, Wonkwang Lee, Myeongjang Pyeon, and Gunhee Kim. Ib-gan: Disengangled representation learning with information bottleneck generative adversarial networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7926–7934, 2021.

Yeonwoo Jeong and Hyun Oh Song. Learning discrete and continuous factors of data via alternating disentanglement. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3091–3099. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/jeong19d.html.

Dimitris Kalatzis, Konstantia Kotta, Ilias Kalamaras, Anastasios Vafeiadis, Andrew Rawstron, Dimitris Tzovaras, and Kostas Stamatopoulos. Towards unsupervised classification with deep generative models, 2018. URL https://openreview.net/forum?id=ryb83alCZ.

T. Anderson Keller and Max Welling. Topographic vaes learn equivariant capsules. *CoRR*, abs/2109.01394, 2021. URL https://arxiv.org/abs/2109.01394.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2649–2658. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/kim18b.html.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013. URL https://arxiv.org/abs/1312.6114.

Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. VARIATIONAL INFERENCE OF DISENTANGLED LATENT CONCEPTS FROM UNLABELED OBSERVATIONS. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1kG7GZAW.

Yann LeCun, Fu Jie Huang, and Léon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR'04, pp. 97–104, USA, 2004. IEEE Computer Society.

Yingzhen Li and Stephan Mandt. Disentangled sequential autoencoder, 2018. URL https://arxiv.org/abs/1803.02991.

Yu-Jhe Li, Ci-Siang Lin, Yan-Bo Lin, and Yu-Chiang Frank 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.

Zhiyuan Li, Jaideep Vitthal Murkute, Prashnna Kumar Gyawali, and Linwei Wang. Progressive learning and disentanglement of hierarchical representations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJxpsxrYPS.

Zinan Lin, Kiran Koshy Thekumparampil, Giulia C. Fanti, and Sewoong Oh. Infogan-cr and modelcentrality: Self-supervised model training and selection for disentangling gans. In *ICML*, pp. 6127–6139, 2020. URL http://proceedings.mlr.press/v119/lin20e.html.

Bingchen Liu, Yizhe Zhu, Zuohui Fu, Gerard de Melo, and Ahmed Elgammal. Oogan: Disentangling gan with one-hot sampling and orthogonal regularization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):4836–4843, Apr. 2020. doi: 10.1609/aaai.v34i04.5919. URL https://ojs.aaai.org/index.php/AAAI/article/view/5919.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4114–4124. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/locatello19a.html.

Emile Mathieu, Tom Rainforth, N Siddharth, and Yee Whye Teh. Disentangling disentanglement in variational autoencoders. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4402–4412. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/mathieu19a.html.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

Ning Miao, Emile Mathieu, N. Siddharth, Yee Whye Teh, and Tom Rainforth. Intel-vaes: Adding inductive biases to variational auto-encoders via intermediary latents, 2021.

Nathan Juraj Michlo. Disent - a modular disentangled representation learning framework for pytorch. Github, 2021. URL https://github.com/nmichlo/disent.

William Peebles, John Peebles, Jun-Yan Zhu, Alexei A. Efros, and Antonio Torralba. The hessian penalty: A weak prior for unsupervised disentanglement. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

Scott E Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. Deep visual analogy-making. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/e07413354875be01a996dc560274708e-Paper.pdf.

Xuanchi Ren, Tao Yang, Yuwang Wang, and Wenjun Zeng. Learning disentangled representation by exploiting pretrained generative models: A contrastive learning view. In *ICLR*, 2022.

Huajie Shao, Shuochao Yao, Dachun Sun, Aston Zhang, Shengzhong Liu, Dongxin Liu, Jun Wang, and Tarek Abdelzaher. ControlVAE: Controllable variational autoencoder. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8655–8664. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/shao20b.html.

Sahil Singla, Besmira Nushi, Shital Shah, Ece Kamar, and Eric 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. URL https://openaccess.thecvf.com/content/CVPR2021/papers/Singla_Understanding_Failures_of_Deep_Networks_via_Robust_Feature_Extraction_CVPR_2021_paper.pdf.

Josh Tenenbaum. Building machines that learn and think like people. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, pp. 5, Richland, SC, 2018. International Foundation for Autonomous Agents and Multiagent Systems.

Nicholas Watters, Loïc Matthey, Christopher P. Burgess, and Alexander Lerchner. Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes. *CoRR*, abs/1901.07017, 2019. URL `http://arxiv.org/abs/1901.07017`.

Changyi Xiao and Ligang Liu. Generative flows with matrix exponential. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10452–10461. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/xiao20a.html`.

Tao Yang, Xuanchi Ren, Yuwang Wang, Wenjun Zeng, Nanning Zheng, and Pengju Ren. Groupifyvae: from group-based definition to vae-based unsupervised representation disentanglement. *CoRR*, abs/2102.10303, 2021. URL `https://arxiv.org/abs/2102.10303`.

Yanchao Yang, Yutong Chen, and Stefano 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.

Xinqi Zhu, Chang Xu, and Dacheng Tao. Commutative lie group VAE for disentanglement learning. *CoRR*, abs/2106.03375, 2021. URL `https://arxiv.org/abs/2106.03375`.

Yang Zou, Xiaodong Yang, Zhiding Yu, Bhagavatula Vijayakumar, and Jan Kautz. Joint disentangling and adaptation for cross-domain person re-identification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

# A Notation

| | | | |
|---|---|---|---|
| $\boldsymbol{z}$ | 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 |
| $Sym_n(\mathbb{R})$ | 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 (equivariant to $G_I$) |
| $\psi_M(\cdot)$ | $\psi_M(\cdot) \in M_n(\mathbb{R})$ | $\psi_{E_M}(\cdot)$ | $\psi_{E_M}(\cdot) \in E_M$ |
| $\psi_{E_S}(\cdot)$ | $\psi_{E_S}(\cdot) \in E_S$ | $\mathbf{0}$ | 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 |
| $\Xi$ | $G_I \times G_L \to G_L$ | $\Gamma$ | $G_L \times G_T \to G_T$ |

# B Objective Function

## B.1 Evidence of Lower Bound (ELBO)

The log likelihood of $p(\boldsymbol{x})$ can be derived as follows:

$$\begin{aligned}
\log p_\theta(\boldsymbol{x}) &= \int q_1(\hat{\boldsymbol{z}}_1|\boldsymbol{x}) q_2(\hat{\boldsymbol{z}}_2|\boldsymbol{x}) \cdots q_k(\hat{\boldsymbol{z}}_k|\boldsymbol{x}) \log p_\theta(\boldsymbol{x}) \mathsf{d}\hat{\boldsymbol{z}}' && (10)\\
&= \int q_1(\hat{\boldsymbol{z}}_1|\boldsymbol{x}) q_2(\hat{\boldsymbol{z}}_2|\boldsymbol{x}) \cdots q_k(\hat{\boldsymbol{z}}_k|\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})} \mathsf{d}\hat{\boldsymbol{z}}' && (11)\\
&= \int q_1(\hat{\boldsymbol{z}}_1|\boldsymbol{x}) q_2(\hat{\boldsymbol{z}}_2|\boldsymbol{x}) \cdots q_k(\hat{\boldsymbol{z}}_k|\boldsymbol{x}) \cdot \\
&\quad \Big[ \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})} \Big] \mathsf{d}\hat{\boldsymbol{z}}' && (12)\\
&\geqq \int q_1(\hat{\boldsymbol{z}}_1|\boldsymbol{x}) q_2(\hat{\boldsymbol{z}}_2|\boldsymbol{x}) \cdots q_k(\hat{\boldsymbol{z}}_k|\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})} \mathsf{d}\hat{\boldsymbol{z}}' && (13)\\
&= \int q_1(\hat{\boldsymbol{z}}_1|\boldsymbol{x}) q_2(\hat{\boldsymbol{z}}_2|\boldsymbol{x}) \cdots q_k(\hat{\boldsymbol{z}}_k|\boldsymbol{x}) \cdot \\
&\quad \Big[ \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})} \Big] \mathsf{d}\hat{\boldsymbol{z}}', && (14)
\end{aligned}$$

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 IE-transformation function operates independently, then $p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \cdots, \hat{\boldsymbol{z}}_k) = \prod_{i=1}^{k} p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_i)$.

$$
\begin{aligned}
\log p_\theta(\boldsymbol{x}) \geq \frac{1}{k}\sum_{i=1}^{k}\left[\int q_i(\hat{\boldsymbol{z}}_i|\boldsymbol{x})\log p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_i)\mathsf{d}\hat{\boldsymbol{z}}_i \prod_{j=1,j\neq i}^{k}\int q_j(\hat{\boldsymbol{z}}_j|\boldsymbol{x})\mathsf{d}\hat{\boldsymbol{z}}_j\right] \\
- \int q_1(\hat{\boldsymbol{z}}_1|\boldsymbol{x})q_2(\hat{\boldsymbol{z}}_2|\boldsymbol{x})\cdots q_k(\hat{\boldsymbol{z}}_k|\boldsymbol{x})\log\frac{q_1(\hat{\boldsymbol{z}}_1|\boldsymbol{x})q_2(\hat{\boldsymbol{z}}_2|\boldsymbol{x})\cdots q_k(\hat{\boldsymbol{z}}_k|\boldsymbol{x})}{p(\hat{\boldsymbol{z}}_1)p(\hat{\boldsymbol{z}}_2)\cdots p(\boldsymbol{z}_k)}\mathsf{d}\hat{\boldsymbol{z}}'
\end{aligned} \tag{15}
$$

$$
= \frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_{q(\hat{\boldsymbol{z}}_i|\boldsymbol{x})}\log p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_i) - \sum_{i=1}^{k}\left[D_{\mathrm{KL}}(q_\phi(\hat{\boldsymbol{z}}_i|\boldsymbol{x})||p(\hat{\boldsymbol{z}}_i))\prod_{j=1,j\neq i}^{k}\int q_j(\hat{\boldsymbol{z}}_j|\boldsymbol{x})\mathsf{d}\hat{\boldsymbol{z}}_j\right] \tag{16}
$$

$$
= \frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_{q_\phi(\hat{\boldsymbol{z}}_i|\boldsymbol{x})}\log p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_i) - \sum_{i=1}^{k}D_{\mathrm{KL}}(q_\phi(\hat{\boldsymbol{z}}_i|\boldsymbol{x})||p(\boldsymbol{z}_i)) \tag{17}
$$

$$
= \frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_{q_{\phi,\psi_i}(\boldsymbol{z}|\boldsymbol{x})}\log p_\theta(\boldsymbol{x}|\psi_i(\boldsymbol{z})) - \sum_{i=1}^{k}D_{\mathrm{KL}}(q_{\phi,\psi_i}(\boldsymbol{z}|\boldsymbol{x})||p_{\psi_i}(\boldsymbol{z})). \tag{18}
$$

Therefore, we define ELBO as:

$$
\mathcal{L}'(\phi,\theta,\psi_{i\in[1,k]};\boldsymbol{x}) = \underbrace{\frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_{q_{\phi,\psi_i}(\boldsymbol{z}_i|\boldsymbol{x})}\log p_\theta(\boldsymbol{x}|\psi_i(\boldsymbol{z}))}_{\text{① reconstruction loss}} - \underbrace{\sum_{i=1}^{k}D_{\mathrm{KL}}(q_{\phi,\psi_i}(\boldsymbol{z}|\boldsymbol{x})||p_{\psi_i}(\boldsymbol{z}))}_{\text{② KL divergence}}. \tag{19}
$$

However, following Eq. 19, 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. 19 represents that increasing number of MIE-transformation is equal to increasing $\beta$ hyper-parameter in $\beta$-VAE (Higgins et al., 2017).

### B.2 Exponential Family Kullback-Leibler Divergence

The second term of Eq. 19 is equal to $D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})||f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}))$ because power density function of posterior and prior are $f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})$ and $f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})$, respectively.

$$
D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})||f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})) = \int_{-\infty}^{\infty}f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\log f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\mathrm{d}\boldsymbol{x} - \int_{-\infty}^{\infty}f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\log f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})\mathrm{d}\boldsymbol{x}. \tag{20}
$$

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

$$
\int_{-\infty}^{\infty}f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\log f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\mathrm{d}\boldsymbol{x} = \int_{-\infty}^{\infty}f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})[\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}^{\intercal}\mathbf{T}(\boldsymbol{x}) - A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) + B(\boldsymbol{x})]\mathrm{d}\boldsymbol{x} \tag{21}
$$

$$
= -A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\cancelto{1}{\int_{-\infty}^{\infty}f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\mathrm{d}\boldsymbol{x}} + \int_{-\infty}^{\infty}f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})[\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}^{\intercal}\mathbf{T}(\boldsymbol{x}) + B(\boldsymbol{x})]\mathrm{d}\boldsymbol{x} \tag{22}
$$

$$
= -A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) + \boldsymbol{\theta}_{\hat{\boldsymbol{z}}}^{\intercal}\int_{-\infty}^{\infty}T(\boldsymbol{x})f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\mathrm{d}\boldsymbol{x} + \int_{-\infty}^{\infty}B(\boldsymbol{x})f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\mathrm{d}\boldsymbol{x}, \tag{23}
$$

and

$$
\int_{-\infty}^{\infty}f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\log f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})\mathrm{d}\boldsymbol{x} = -A(\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}) + \boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}^{\intercal}\int_{-\infty}^{\infty}T(\boldsymbol{x})f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})\mathrm{d}\boldsymbol{x} + \int_{-\infty}^{\infty}B(\boldsymbol{x})f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})\mathrm{d}\boldsymbol{x}. \tag{24}
$$

$$
\begin{aligned}
\therefore D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})||f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})) = A(\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}) - A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) + \boldsymbol{\theta}_{\hat{\boldsymbol{z}}}\int_{-\infty}^{\infty}T(\boldsymbol{x})f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\mathrm{d}\boldsymbol{x} \\
- \boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}\int_{-\infty}^{\infty}T(\boldsymbol{x})f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})\mathrm{d}\boldsymbol{x}.
\end{aligned} \tag{25}
$$

The mean of the sufficient statistic is followed as:

$$\int_{-\infty}^{\infty} T(\boldsymbol{x}) f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta})\mathrm{d}\boldsymbol{x} = \frac{\partial A^*(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \approx \frac{\partial A(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \qquad \because A^*(\boldsymbol{\theta}) = \boldsymbol{\theta}^\intercal A^*, \tag{26}$$

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 (Chen et al., 2018). Then, we approximate $A^*$ to $A$, and train $A$ to be close to $A^*$. Consequently, we obtain KL divergence of the exponential family as:

$$\int_{-\infty}^{\infty} f_{\boldsymbol{x}}(\mathrm{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) \log f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) = -A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) + \boldsymbol{\theta}_{\hat{\boldsymbol{z}}}^\intercal \frac{\partial A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})}{\partial \boldsymbol{\theta}_{\hat{\boldsymbol{z}}}} + \int_{-\infty}^{\infty} f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) B(\boldsymbol{x})\mathrm{d}\boldsymbol{x}, \tag{27}$$

$$\int_{-\infty}^{\infty} f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) \log f_{\boldsymbol{x}}(\mathrm{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}) = -Z(\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}) + \boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}^\intercal \frac{\partial A(\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})}{\partial \boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}} + \int_{-\infty}^{\infty} f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) B(\boldsymbol{x})\mathrm{d}\boldsymbol{x}. \tag{28}$$

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

$$D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})||f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})) = A(\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}) - A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) + \boldsymbol{\theta}_{\hat{\boldsymbol{z}}}^\intercal \frac{\partial A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})}{\partial \boldsymbol{\theta}_{\hat{\boldsymbol{z}}}} - \boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}^\intercal \frac{\partial A(\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})}{\partial \boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}}. \tag{29}$$

### B.3 Backgroud of Exponential Family

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

$$\begin{aligned} f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}) &= h(\boldsymbol{x})\mathrm{exp}(\boldsymbol{\theta}^\intercal T(\boldsymbol{x}) - A(\boldsymbol{\theta})) \\ &= \mathrm{exp}(\boldsymbol{\theta}^\intercal T(\boldsymbol{x}) - A(\boldsymbol{\theta}) + B(\boldsymbol{x})), \end{aligned} \tag{30}$$

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* $\boldsymbol{\theta}$. However, we set $T(\cdot)$, $A(\cdot)$, and $B(\cdot)$ are deterministic functions by maximizing conjugate prior for parameter $\xi$. To determine the *natural parameter* of posterior and prior $\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m}$, and $\hat{\boldsymbol{\epsilon}}_m$, we use a natural parameter generator (NPG) designed by multi-layer perceptron (Charpentier et al., 2022a). As introduced in Bishop (2006); Charpentier et al. (2022a), we assume exponential family always admits a conjugate prior:

$$q(\boldsymbol{\theta}|\xi, \boldsymbol{\nu}) = \mathrm{exp}(\boldsymbol{\nu}\boldsymbol{\theta}^\intercal \xi - \boldsymbol{\nu} A(\boldsymbol{\theta}) + B'(\xi, \boldsymbol{\nu})), \tag{31}$$

where $B'(\cdot)$ is a *normalize coefficient* and $\nu$ is evidence, and it is expressed by prior natural parameter $\xi$. 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. 30 is modified: $p(\mathbf{X}|\boldsymbol{\theta}) = \prod_{n=1}^{N} h(\mathrm{x}_n)\mathrm{exp}(\boldsymbol{\theta}^\intercal \sum_{n=1}^{N} T(\mathrm{x}_n) - A(\boldsymbol{\theta}))$ (Bishop, 2006), where observation $\mathbf{X} = \{\mathrm{x}_1, \cdots \mathrm{x}_N\}$. In the next, we multiply the modified formation by the prior Eq. 31 to obtain the posterior distribution (Bishop, 2006) as Eq. 1.

## C Proof

### C.1 $E_M \not\subset E_S$

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 $\boldsymbol{M} =$

$\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}$,

$$\begin{aligned}
\mathbf{e}^{\boldsymbol{M}} &= \sum_{k=0}^{\infty} \frac{1}{k!}\boldsymbol{M}^k \\
&= I + \begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix} + \frac{1}{2!}\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}^2 + \cdots + \frac{1}{(n-1)!}\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}^{(n-1)} + \cdots \\
&= I + \begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix} + \frac{1}{2!}\begin{bmatrix}1 & 2\\ 0 & 1\end{bmatrix} + \ldots + \frac{1}{(n-1)!}\begin{bmatrix}1 & n-1\\ 0 & 1\end{bmatrix} + \cdots \\
&= I + \begin{bmatrix}\sum_{n=0}^{\infty}\frac{1}{n!} & 1+\sum_{n=0}^{\infty}\frac{1}{(n-1)!}\\ 0 & \sum_{n=0}^{\infty}\frac{1}{n!}\end{bmatrix} \\
&= \begin{bmatrix}1+\sum_{n=0}^{\infty}\frac{1}{n!} & 1+\sum_{n=0}^{\infty}\frac{1}{(n-1)!}\\ 0 & 1+\sum_{n=0}^{\infty}\frac{1}{n!}\end{bmatrix} \\
&= \begin{bmatrix}1+e & 1+e\\ 0 & 1+e\end{bmatrix}.
\end{aligned} \tag{32}$$

The matrix $\mathbf{e}^{\boldsymbol{M}}$ is asymmetric and not in $E_S$. Therefore $E_M \not\subset E_S$.

### C.2 Transitivity of Equivariance

We show that $\psi_{G_S}$ preserves equivariance between group elements of latent space in @ and input space. If there exists equivariance 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 ($q_\phi(g_I x) = g_L q_\phi(x)$).

Let $g_L^a$ be a group element in @, and $g_I^a$ is a group element in $G_I$ corresponding to @, and $g_T^a$ is a group element where corresponding to @ on the latent vector space transformed from the original latent vector space. Then, group element $g_T^a$ is equal to $g_L^a$:

$$\hat{\boldsymbol{z}}_1 = \psi_{G_S}(\boldsymbol{z}_1), \text{and} \tag{33}$$

$$\hat{\boldsymbol{z}}_2 = \psi_{G_S}(\boldsymbol{z}_2) = \psi_{G_S}(g_L^a \boldsymbol{z}_1) = g_L^a \psi_{G_S}(\boldsymbol{z}_1) \; (\because \text{Proposition 1}), \tag{34}$$

$$\text{then } g_L^a \psi_{G_S}(\boldsymbol{z}_1) = g_T^a \psi_{G_S}(\boldsymbol{z}_1) \; (\because \hat{\boldsymbol{z}}_2 = g_T^a \hat{\boldsymbol{z}}_1) \tag{35}$$

$$\Rightarrow (g_L^a - g_T^a)\psi_{G_S}(\boldsymbol{z}_1) = \mathbf{0}, \tag{36}$$

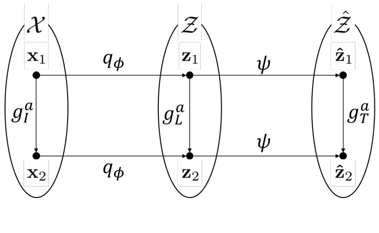


Figure 5: Equivariant map: $\mathcal{X}$, $\mathcal{Z}$, and $\hat{\mathcal{Z}}$ are input space, latent vector space, and transformed latent vector space by L2L transfomration function $\psi(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$. respectively. $\boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{z} \in \mathcal{Z}$, and $\hat{\boldsymbol{z}} \in \hat{\mathcal{Z}}$.

where **0** is a zero vector. Eq. 35 is defined when $\forall \boldsymbol{z} \in \mathcal{Z}$ by the equivariance definition. In other words, Eq. 35 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 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 @ if the latent vector $\boldsymbol{z}$ is transformed by $\psi_{G_S}$.

## D Model and Training Procedure Details

### D.1 Hyper-Parameters

We set 256 mini-batch size in the datasets (dSprites, 3D Shapes, and 3D Cars), Adam optimizer with learning rate $4 \times 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and epochs from $\{30, 67, 200\}$ as a common setting

| models | hyper-parameters | values |
|---|---|---|
| common | batch size | 256 |
| | epoch | {30, 200} |
| | optim | Adam |
| | lr | 4e-4 |
| | lr for MIET | 4e-4 |
| | weight decay | 1e-4 |
| | latent dim | 10 |
| $\beta$-VAE | # of IE and EF | {1, 2, 4, 10} |
| $\beta$-TCVAE | $\beta$ | {4, 6} |
| | # of IE and EF | {1, 3} |
| | $\alpha$, $\gamma$ | 1.0 |
| commut-VAE | $\lambda_{\text{decomp}}$ | 40 |
| | $\lambda_{\text{hessian}}$ | 40 |
| | forward group | 0.2 |
| | group reconst | {0.2, 0.5, 0.7} |

(a) dSprites and 3D Cars: epochs for dSprites and 3D cars are 30 and 200, respectively.

| models | hyper-parameters | values |
|---|---|---|
| common | batch size | 256 |
| | epoch | 67 |
| | optim | Adam |
| | lr | 4e-4 |
| | lr for MIET | 4e-4 |
| $\beta$-VAE | # of IE and EF | {1, 2, 4, 10} |
| | weight decay | 0.0 |
| | latent dim | 6 |
| $\beta$-TCVAE | $\beta$ | {4, 6} |
| | # of IE and EF | {1, 3} |
| | $\alpha$, $\gamma$ | 1.0 |
| | weight decay | 1e-4 |
| | latent dim | 6 |
| commut-VAE | $\lambda_{\text{decomp}}$ | 40 |
| | $\lambda_{\text{hessian}}$ | 40 |
| | forward group | 0.2 |
| | group reconst | {0.2, 0.5, 0.7} |
| | weight decay | 0.0 |
| | latent dim | 10 |

(b) 3D Shapes

Table 7: Notation commut-VAE is CLG-VAEs, lr is learning rate, latent dim is dimension size of latent vector, group reconst is group reconstrunction, and forward group is forward group pass.

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 Kim & Mnih (2018); Ren et al. (2022). We tune $\beta$ from $\{1, 2, 4, 10\}$ and $\{4, 6\}$ for $\beta$-VAE and $\beta$-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 Zhu et al. (2021). Regarding the CLG-VAE, we fix $\lambda_{\text{decomp}}$, $\lambda_{\text{hessian}}$, and forward group features as 40, 20, and 0.2, respectively. Because the hyper-parameters showed the best result in Zhu et al. (2021). We set group reconstruction from $\{0.2, 0.5, 0.7\}$. In addition, we set masking ratio $\lambda$ from $\{0.0, 0.5, \ldots 2.0, \infty\}$. To check the impact of MIE-transformation, we do not consider the Groupified VAE because the latter is implemented with an extended decoder (different capacity).

### D.2 Training Procedure and Model

We present the model architecture for each dataset in Table 8 and 9. For MIET architecture reproduction, Algorithm 2 to Algorithm 4. To estimate EF similarity loss, we directly implement $\nabla_{\hat{\boldsymbol{z}},\hat{\boldsymbol{\epsilon}},\boldsymbol{\lambda}_m} p(\boldsymbol{\theta}|\mathbf{X},\mathcal{X},\boldsymbol{\nu}) + \boldsymbol{\lambda}_m D_{\text{KL}}(f_{\hat{\boldsymbol{z}}}(\hat{z}|\theta_{\hat{\boldsymbol{z}}})||f_{\hat{\boldsymbol{\epsilon}}}(\hat{\epsilon}|\theta_{\hat{\boldsymbol{\epsilon}}}))$, instead of implementing posterior, in our supplement materials. More details of this algorithm are in `ie_transformation.py` file.

| Encoder | Decoder |
|---|---|
| Input 64 × 64 binary image | input $\in \mathbb{R}^{10}$ |
| 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 8: VAE architecture for dSprites dataset.

**Algorithm 1** Unit Invertible and Equivariant Transformation Function (UIET-function)

```
Input: matrices $\boldsymbol{M}_1$, and $\boldsymbol{M}_2$
Output: Invertible and Equivariant Transformation Function $\psi(\cdot)$
  $\boldsymbol{M}_1, \boldsymbol{M}_2 \leftarrow \frac{1}{2}(\boldsymbol{M}_1 + \boldsymbol{M}_1^\intercal), \frac{1}{2}(\boldsymbol{M}_2 + \boldsymbol{M}_2^\intercal)$
  $\psi(\cdot) \leftarrow \boldsymbol{M}_1^\intercal \boldsymbol{M}_2$
```

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

Table 9: VAE architecture for 3D Shapes, and 3D Cars datasets. For exceptional case, CLG-VAE, we ues ten dimension size on 3D Shapes dataset (Zhu et al., 2021).

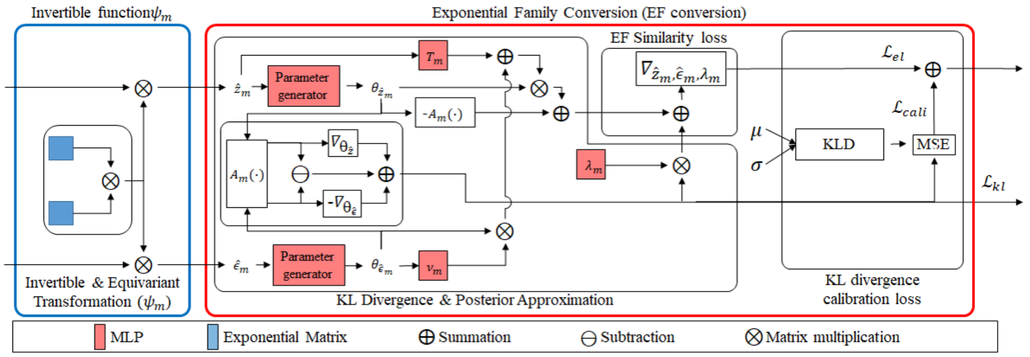


Figure 6: The overall architecture of our proposed *MIET*-VAE. KLD is the KL divergence of Gaussian Distribution in Kingma & Welling (2013). $\nabla_{\hat{\boldsymbol{z}}_m,\hat{\boldsymbol{\epsilon}}_m,\boldsymbol{\lambda}_m}$ is in Eq. 4, and MSE is a mean squared error. More details of our proposed method are introduced in Section 3.

**Algorithm 2** IE-Transformation

```
Input: latent vector $\boldsymbol{z}$, and samples from prior $\boldsymbol{\epsilon}$
Output: 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

```
Input: 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 $\boldsymbol{\nu}$.
Output: KL divergence $D_{\mathrm{KL}}(f_{\hat{\boldsymbol{z}}}(\hat{z}|\theta_{\hat{\boldsymbol{z}}})||f_{\hat{\boldsymbol{\epsilon}}}(\hat{\epsilon}|\theta_{\hat{\boldsymbol{\epsilon}}}))$, and posterior $p(\boldsymbol{\theta}|\mathbf{X},\mathcal{X},\boldsymbol{\nu})$
  $\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}, \boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}} \leftarrow \Omega_1(\hat{z}), \Omega_2(\hat{\epsilon})$
  $A \leftarrow$ sparse log-normalizer($A$)                                        ▷ Equation 7
  $p(\boldsymbol{\theta}|\mathbf{X},\mathcal{X},\boldsymbol{\nu}) \leftarrow \exp\Big[\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}(\sum_{i=1}^{B} T(\hat{z}_i) + \boldsymbol{\nu}\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}) - A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\Big]$          ▷ Equation 1
  $D_{\mathrm{KL}}(f_{\hat{\boldsymbol{z}}}(\hat{z}|\theta_{\hat{\boldsymbol{z}}})||f_{\hat{\boldsymbol{\epsilon}}}(\hat{\epsilon}|\theta_{\hat{\boldsymbol{\epsilon}}})) \leftarrow A(\boldsymbol{\theta}_{\hat{\epsilon}}) - A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) + \boldsymbol{\theta}_{\hat{\boldsymbol{z}}}^{\intercal}\frac{\partial A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})}{\partial \boldsymbol{\theta}_{\hat{\boldsymbol{z}}}} - \boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}^{\intercal}\frac{\partial A(\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})}{\partial \boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}}$     ▷ Equation 5
```

**Algorithm 4** EF-Conversion Loss

```
Input: KL divergence $D_{\mathrm{KL}}(f_{\hat{\boldsymbol{z}}}(\hat{z}|\theta_{\hat{\boldsymbol{z}}})||f_{\hat{\boldsymbol{\epsilon}}}(\hat{\epsilon}|\theta_{\hat{\boldsymbol{\epsilon}}}))$, posterior $p(\boldsymbol{\theta}|\mathbf{X},\mathcal{X},\boldsymbol{\nu})$, $\mu$,$\sigma$
Output: Regularization $\mathcal{L}_{reg}$
  $\mathcal{L}_{el} \leftarrow p(\boldsymbol{\theta}|\mathbf{X},\mathcal{X},\boldsymbol{\nu}) + \boldsymbol{\lambda}_m D_{\mathrm{KL}}(f_{\hat{\boldsymbol{z}}}(\hat{z}|\theta_{\hat{\boldsymbol{z}}})||f_{\hat{\boldsymbol{\epsilon}}}(\hat{\epsilon}|\theta_{\hat{\boldsymbol{\epsilon}}}))$
  $\mathcal{L}_{el} \leftarrow ||\nabla_{\hat{\boldsymbol{z}}_m,\hat{\boldsymbol{\epsilon}}_m,\boldsymbol{\lambda}_m}\mathcal{L}_{el}||_2^2$
  $D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})) \leftarrow 0.5\sum_{d=1}^{D}(1 + 2\log\sigma_j - \mu_j^2 - \sigma_j^2)$     ▷ Reference (Kingma & Welling, 2013)
  $\mathcal{L}_{cali} \leftarrow$ MSE($D_{\mathrm{KL}}(f_{\hat{\boldsymbol{z}}}(\hat{z}|\theta_{\hat{\boldsymbol{z}}})||f_{\hat{\boldsymbol{\epsilon}}}(\hat{\epsilon}|\theta_{\hat{\boldsymbol{\epsilon}}}))$, $D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}))$)     ▷ Equation 6
  $\mathcal{L} \leftarrow \mathcal{L}_{el} + \mathcal{L}_{cali}$                                     ▷ Equation 9
```

# E Dataset Details

We compare well-known VAEs to CHIC-VAEs: VAE, $\beta$-VAE, $\beta$-TCVAE, and CLG-VAE on the following data sets with 1) dSprites (Matthey et al., 2017) which consists of 737,280 binary $64 \times 64$ 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 (Burgess & Kim, 2018) 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 object color(10). 3) 3D Cars (Reed et al., 2015) 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).

# F Quantitative Analysis Setting

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 Kim & Mnih (2018). For the other metrics, we follow default values introduced in Michlo (2021), training and evaluation 100 and 50 times with 100 mini-batches, respectively.

# G Quantitative Analysis with more Baselines and Datasets

We investigate more baseline and dataset. We add Factor-VAE (Kim & Mnih, 2018) and Control-VAE (Shao et al., 2020) model, and smallNORB (LeCun et al., 2004) dataset. We represent additional results in Table 10. The proposed method improves the disentanglement performance on Control-VAE, and Factor-VAE. Also, in the large-scale task, smallNORB ($96 \times 96$), our method shows better results than the original cases.

| dSprites | Disentanglement Metric | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | FVM ↑ | | MIG ↑ | | SAP ↑ | | DCI ↑ | |
| | original | MIET | original | MIET | original | MIET | original | MIET |
| Factor-VAE | 59.28(±6.93) | **59.55**(±5.62) | 3.30(±1.97) | **3.64**(±1.42) | **1.09**(±0.84) | 0.92(±0.56) | 10.46(±2.76) | **11.49**(±3.15) |
| Control-VAE | 62.36(±8.62) | **67.71**(±6.41) | 4.36(±2.86) | **7.34**(±4.10) | **2.11**(±1.88) | 1.93(±1.63) | 10.40(±3.42) | **15.18**(±4.61) |

| 3D Shapes | Disentanglement Metric | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | FVM ↑ | | MIG ↑ | | SAP ↑ | | DCI ↑ | |
| | original | MIET | original | MIET | original | MIET | original | MIET |
| Factor-VAE | 41.86(±6.90) | **49.22**(±10.58) | 2.47(±1.88) | **7.47**(±9.48) | 1.18(±0.97) | **2.04**(±1.96) | 10.40(±4.20) | **13.08**(±10.24) |
| Control-VAE | 71.05(±14.35) | **71.89**(±8.33) | 24.88(±13.68) | **32.28**(±10.74) | 6.60(±3.59) | **7.14**(±2.09) | 40.08(±13.45) | **43.06**(±8.68) |

| 3D Cars | Disentanglement Metric | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | FVM ↑ | | MIG ↑ | | SAP ↑ | | DCI ↑ | |
| | original | MIET | original | MIET | original | MIET | original | MIET |
| Factor-VAE | 82.81(±8.32) | **84.84**(±9.30) | 1.22(±0.68) | **2.00**(±1.40) | 0.68(±0.52) | **0.85**(±0.68) | **11.67**(±4.74) | 9.63(±2.72) |
| control-VAE | 88.76(±7.66) | **89.10**(±6.90) | 4.68(±2.67) | **5.08**(±2.68) | 1.16(±0.74) | **1.45**(±0.86) | 14.70(±3.84) | **15.22**(±4.15) |

| smallNORB<br>96×96 | Disentanglement Metric | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | FVM ↑ | | MIG ↑ | | SAP ↑ | | DCI ↑ | |
| | original | MIET | original | MIET | original | MIET | original | MIET |
| CLG-VAE | 32.01(±4.72) | **32.71**(±5.59) | **17.30**(±0.75) | 16.75(±2.57) | 8.82(±0.60) | **9.26**(±1.39) | 11.92(±1.75) | **13.26**(±2.03) |
| Control-VAE | 34.41(±7.17) | **34.46**(±8.64) | 17.37(±1.74) | **17.42**(±1.92) | 9.28(±1.02) | **9.74**(±1.20) | 16.73(±2.70) | **17.76**(±2.84) |

Table 10: The smallNORB $64 \times 64$ represents the centered cropped version of smallNORB. We set the hyper-parameter $\gamma$ of Factor-VAE from {5, 10, 15}, and run total 30 seeds. And we set the maximum KL divergence value of Control-VAE from {10, 12, 14, 16, 18, 20}, and run total 60 seeds.

## G.1 Additional Dataset

The smallNORB (LeCun et al., 2004) dataset consists of total $96 \times 96$ 24,300 grayscale images with five factors, which are category(5), elevation(9), azimuth(18), light(6), right-left(2).

# H Sensitivity to the number of IE-transformation

We present the sensitivity of the number of IE-transformation results of 3D Shapes and 3D Cars in Table 11. These results also show that the disentanglement performance of multiple units of IE-transformation is higher than a single unit.

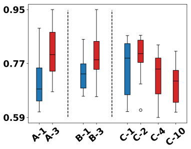
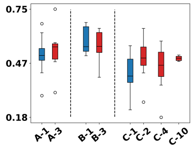
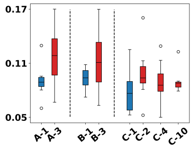
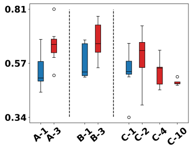
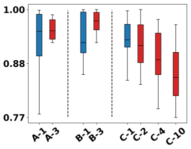
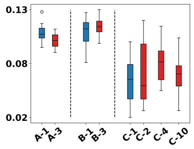
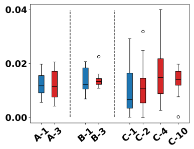
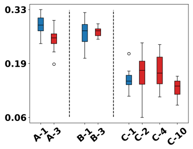
| dataset | Metrics | | | |
|---|---|---|---|---|
| | FVM | IMG | SAP | DCI |
| 3D Shapes |  |  |  |  |
| 3D Cars |  |  |  |  |

Table 11: Impact of the number of MIE-transformation function on the $\beta$-TCVAE and $\beta$-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 IE-transformation cases, respectively. (A-$n$: MIET-$\beta$-TCVAE (4), B-$n$: MIET-$\beta$-TCVAE (6), C-$n$: MIET-$\beta$-VAE, $n$: the number of MIE-transformation)

## I Ablation study

Ablation studies with dSprites and 3D Shapes datasets are presented in Table 12.

| dSprites | $\beta$-VAE | | | $\beta$-TCVAE | | |
|---|---|---|---|---|---|---|
| | MIET | MIET (w/o E) | MIET (w/o EF) | MIET | MIET (w/o E) | MIET (w/o EF) |
| FVM | **74.19**($\pm$5.62) | 71.54($\pm$8.66) | 25.83($\pm$1.16) | **79.87**($\pm$5.80) | 76.39($\pm$7.44) | 77.44($\pm$7.15) |
| MIG | **19.72**($\pm$11.37) | 19.29($\pm$11.79) | 0.02($\pm$0.01) | **35.04**($\pm$4.07) | 33.83($\pm$8.06) | 21.88($\pm$8.42) |
| SAP | **5.08**($\pm$2.90) | 4.91($\pm$3.25) | 0.21($\pm$0.10) | **7.70**($\pm$1.63) | 7.64($\pm$2.03) | 6.84($\pm$1.87) |
| DCI | **28.81**($\pm$10.19) | 27.51($\pm$11.49) | 1.81($\pm$0.08) | **47.83**($\pm$5.01) | 45.10($\pm$6.92) | 37.84($\pm$8.85) |

| 3D Shapes | $\beta$-VAE | | | $\beta$-TCVAE | | |
|---|---|---|---|---|---|---|
| | MIET | MIET (w/o E) | MIET (w/o EF) | MIET | MIET (w/o E) | MIET (w/o EF) |
| FVM | **75.19**($\pm$8.16) | 74.91($\pm$10.46) | 22.27($\pm$1.29) | **80.59**($\pm$8.57) | 77.90($\pm$8.66) | 66.38($\pm$7.57) |
| MIG | 47.37($\pm$10.13) | **47.45**($\pm$8.98) | 0.28($\pm$0.09) | **54.49**($\pm$9.44) | 51.37($\pm$11.54) | 36.08($\pm$17.42) |
| SAP | 9.20($\pm$2.44) | **9.43**($\pm$2.59) | 0.26($\pm$0.07) | **11.58**($\pm$3.32) | 10.23($\pm$3.13) | 7.13($\pm$3.09) |
| DCI | **54.95**($\pm$8.99) | 54.23($\pm$9.05) | 0.10($\pm$0.02) | **66.22**($\pm$7.32) | 61.18($\pm$8.87) | 56.85($\pm$11.72) |

Table 12: Ablation study of the impact of the equivariant property (w/o E), and EF-conversion (w/o EF). We conduct 40 $\beta$-VAE models and 20 $\beta$-TCVAE with different hyper-parameters on the four disentanglement metrics.

## J Additional Experiment of Computing Complexity

We additionally estimate the computing complexity depending on the number of IE-transformation. The results are in Table 13 and represent the training time complexity compare to baselines (when the number of IE is equal to 0).

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

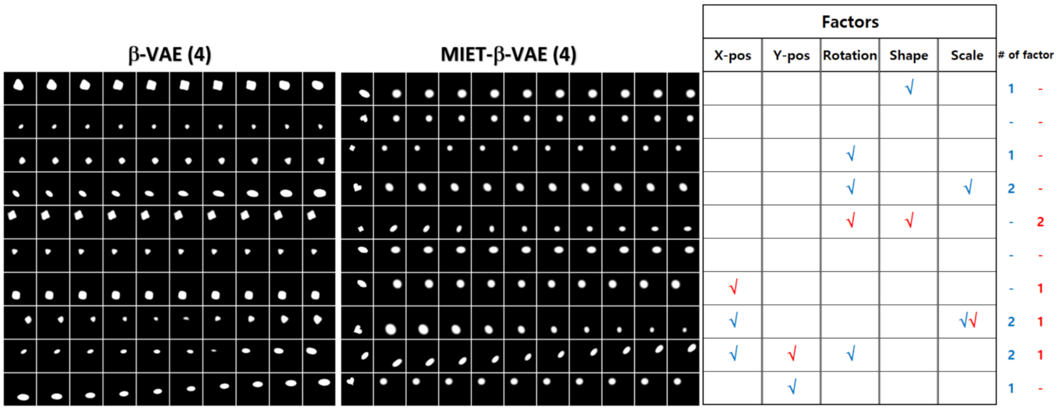
Table 13: Training computing complexity.

## K Qualitative Analysis

We represent more qualitative analysis with all datasets in Fig 7-11.

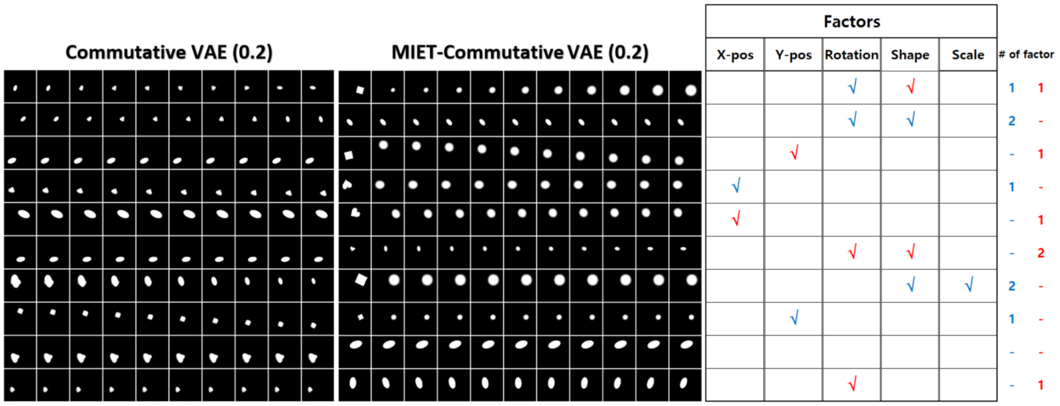
## L Additional Inductive Bias

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), Li & Mandt (2018); Bai et al. (2021) 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



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

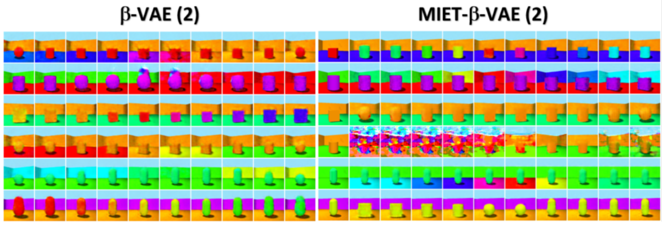
Figure 7: Qualitative analysis result of $\beta$-VAE and MIET-$\beta$-VAE. As shown in the figure, $\beta$-VAE struggles with rotation and scale factors in $4^{th}$dimension. Also, it struggles with x-position and scale factors in $8^{th}$ dimension, and x-position and rotation factors in $9^{th}$ dimension. However, MIET-$\beta$-VAE only struggles with rotation and shape factors in $5^{th}$ dimension.



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

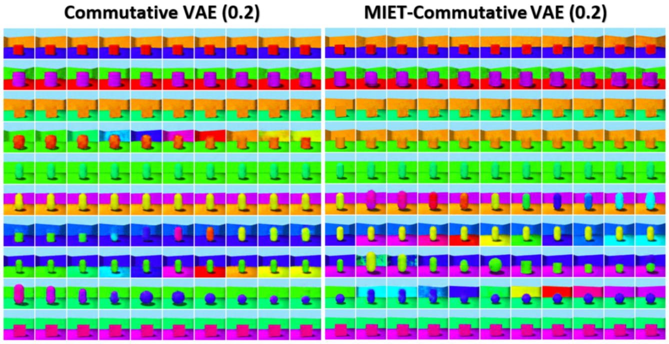
Figure 8: Qualitative analysis result of CLG-VAE (0.2) and MIET-CLG-VAE (0.2) with dSprites. As shown in the results, CLG-VAE struggles with rotation and shape factors in $2^{nd}$ dimension, and shape and scale factors in $7^{th}$ dimension. However, MIET-CLG-VAE separates rotation and shape factors in $10^{th}$, and $1^{st}$ dimensions respectively.

$f$. Moreover Bai et al. (2021) 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 the equivariant function between the input and latent vector space.



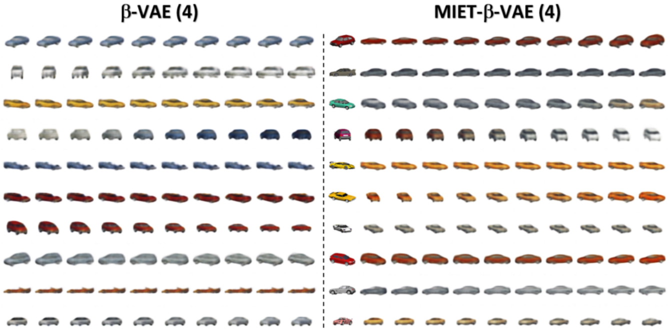
| Factors | | | | | | |
|---|---|---|---|---|---|---|
| Shape | Orien | Scale | Wall | Floor | Object | # of factor |
| √√ | | | | √ | √ | 2 2 |
| √√ | | | √√ | | | 2 2 |
| √ | | √ | | | √ | 2 1 |
| √√ | √ | | | √ | | 2 2 |
| √ | √ | | √ | √ | | 3 1 |
| √√ | | √ | | | √ | 1 2 |

Figure 9: 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. As shown in the results, $beta$-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 MIET-$\beta$-VAE struggles with reconstruction, it is less struggle with than $\beta$-VAE.



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

Figure 10: Shape is object shape, Orien is an orientation of object, Scale is scale factor of object, Wall is wall color factor, Floor is floor color, and Object is object color factor. As shown in the result, 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, MIET-CLG-VAE separates shape, wall, and object color factors.



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

Figure 11: 3D Cars: On the left side is the $\beta$-TCVAE result, and it struggles with body, and azimuth factors shown in the $7^{th}$ row. However, MIET-$\beta$-TCVAE separates azimuth ($6^{th}$ row) and body ($1^{st}$ row). In particular, MIET-$\beta$-TCVAE learns *color* factor ($3^{rd}$ row) which does not exist on $\beta$-TCVAE.