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

Hee-Jun Jung
Gwangju Institute of Science and Technology

Jaehyoung Jeong
Gwangju Institute of Science and Technology

Kangil Kim*
Gwangju Institute of Science and Technology

Abstract

Disentanglement learning is central to understanding and reusing learned representations in variational autoencoders (VAEs). Although equivariance has been explored in this context, effectively exploiting it for disentanglement remains challenging. In this paper, we propose a novel method, called *Multiple Invertible and Partial-Equivariant Transformation* (MIPE-Transformation), which integrates two main parts: (1) *Invertible and Partial-Equivariant Transformation* (IPE-Transformation), guaranteeing an invertible latent-to–transformed-latent mapping while preserving partial input-to-latent equivariance in the transformed latent space; and (2) *Exponential-Family Conversion* (EF-Conversion) to extend the standard Gaussian prior to an approximate exponential family via a learnable conversion. In experiments on the 3D Cars, 3D Shapes, and dSprites datasets, MIPE-Transformation improves the disentanglement performance of state-of-the-art VAEs.

1 Introduction

Disentanglement learning aims to learn interpretable representations and has utility in tasks such as classification (Singla et al., 2021), zero-shot learning (Tenenbaum, 2018), and domain adaptation (Li et al., 2019; Zou et al., 2020). A disentangled representation varies along one latent coordinate in a way that corresponds to a single factor of variation while keeping other factors essentially fixed. Among existing approaches, the Variational Autoencoder (VAE) (Kingma and Welling, 2013) is a primary framework; unsupervised methods encourage factorized latents and reduce inter-dimension correlation (Chen et al., 2018; Higgins et al., 2017; Jeong and Song, 2019; Kim and Mnih, 2018; Li et al., 2020), but Locatello et al. (2019) shows that disentanglement is not identifiable without inductive bias.

To inject such bias, prior work follows two lines: 1) statistics-driven methods that replace the standard Gaussian prior with richer, often non-Gaussian alternatives. For example, Kim et al. (2019); Mathieu et al. (2018) investigate Gaussian-mixture and Student-t priors and demonstrate limitations of the standard Gaussian prior for disentanglement. Similarly, Locatello et al. (2020); Togo et al. (2025) consider priors beyond the Gaussian family and report improved disentanglement performance. However, these approaches are grounded in the original notion of disentangled representation (Bengio et al., 2013) rather than the group-theoretic formulation, and they do not explicitly encode equivariance.

The second line comprises group-theoretic methods that recast disentanglement in terms of the direct product of groups defined on the latent space (Higgins et al., 2018) by equivariant modeling between input and latent vector space rather than 2D planar symmetries. Representative works instantiate the general linear group ($GL(n)$) (Jung et al., 2024; Winter et al., 2022; Zhu et al., 2021) or cyclic groups (Keurti et al., 2023; Yang et al., 2022) on latent vector space, and design equivariant VAE objectives that improve disentangled representation quality. However, these models typically retain a fixed standard Gaussian prior, which limits statistical flexibility.

Motivated by these observations, we target two desiderata: 1) preserving partial equivariance between the input and a transformed latent space, and 2) enabling diverse priors. We introduce a learnable, invertible latent-to-latent transformation that preserves partial equivariance between inputs and latent space, and

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. * Corresponding Author.

via an *exponential-family conversion* (EF-conversion), maps the transformed latent vectors to flexible (non-Gaussian) priors. We refer to the module as the Multiple Invertible and Partial-Equivariant (MIPE) transformation, which can be inserted into existing VAEs.

Through quantitative and qualitative experiments on 3D Cars, 3D Shapes, and dSprites, MIPE yields consistent improvements in disentangled representation learning. Our contributions are:

1. A symmetric matrix exponential latent-to-latent transformation with invertibility and analyzed partial equivariance properties.
2. A training procedure and losses that allow VAEs to learn an unknown latent distribution approximated by an exponential family.
3. The MIPE architecture that integrates multiple IPE transformations with EF-conversion and can be plugged into state-of-the-art VAEs.
4. Empirical analysis demonstrating the effectiveness of MIPE on standard disentanglement benchmarks.

2 Preliminaries

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

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

Group action: Let set X, and $(G, \circ)$ be a group, binary operation $\cdot : G \times X \to X$, then group action $\alpha : \alpha(g, x) = g \cdot x$ following properties:

- Identity: $e \cdot x = x$, where $e \in G,\ x \in X$.
- Compatibility: $\forall g_1, g_2 \in G,\ x \in X,\ \alpha((g_1 \circ g_2), x) = \alpha(g_1, \alpha(g_2, x))$.

Equivariant map: Given X and Y are G-set, and group action $\rho : G \times Y \to Y$. Then a function $f : X \to Y$ is equivariant if

$$f(\alpha(g, x)) = \rho(g, f(x)). \tag{1}$$

Partial-Equivariant map: Let subset of G be $G' \subset G$, then a function $f : X \to Y$ is a partial-equivariant if

$$f(\alpha(g', x)) = \rho(g', f(x)) \text{ where } g' \in G'. \tag{2}$$

Homogeneous space: Let a group G act (on the left) on a smooth manifold X via $\cdot : G \times X \to X$. The G-space X is called *homogeneous* if the action is *transitive*, i.e., for all $x, y \in X$ there exists $g \in G$ with $g \cdot x = y$.

Exponential Family We introduce the conjugate prior and posterior distribution of the exponential family.

Power Density Function:

$$P\boldsymbol{x}(\boldsymbol{x}|\boldsymbol{\theta}) = \exp(\boldsymbol{\theta}^\intercal T(\boldsymbol{x}) - A(\boldsymbol{\theta}) + B(\boldsymbol{x})), \tag{3}$$

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* $\boldsymbol{\theta}$.

If observation is a set of independent identically distributed, then Eq. 3 is modified as

$$p(\mathbf{X}|\boldsymbol{\theta}) = \prod_{n=1}^{N} h(\boldsymbol{x}_n) \exp(\boldsymbol{\theta}^\intercal \sum_{n=1}^{N} T(\boldsymbol{x}_n) - A(\boldsymbol{\theta})), \tag{4}$$

where observation $\mathbf{X} = \{\boldsymbol{x}_1, \cdots \boldsymbol{x}_N\}$ (Bishop, 2006).

Conjugate Prior: Exponential family admits a conjugate prior:

$$q(\boldsymbol{\theta}|\xi, \boldsymbol{\nu}) = \exp(\boldsymbol{\nu}\boldsymbol{\theta}^\intercal \xi - \boldsymbol{\nu} A(\boldsymbol{\theta}) + B'(\xi, \boldsymbol{\nu})), \tag{5}$$

where $B'(\cdot)$ is a *normalize coefficient* and ν is evidence, and it is expressed by prior natural parameter ξ.

Posterior: Let multiply the prior Eq. 5 by the Eq. 4 to obtain the posterior distribution in the form

$$p(\boldsymbol{\theta}|\mathbf{X}, \xi, \boldsymbol{\nu}) \propto \exp(\boldsymbol{\theta}^\intercal (\sum_{n=1}^{N} T(\boldsymbol{x}_n) + \boldsymbol{\nu}\xi) - A(\boldsymbol{\theta})), \tag{6}$$

where the *evidence* $\boldsymbol{\nu}$, *sufficient statistics* $T(\cdot)$ and *log-normalizer*, $A(\cdot)$ are known functions, samples $\mathbf{X}$ from distribution, and *natural parameter* of posterior $\boldsymbol{\theta}$ and of prior ξ (Bishop, 2006).

3 Related Work

3.1 Statistical Approaches for Disentangled Representations

Recently, various studies have focused on unsupervised disentanglement learning. β-VAE (Higgins et al., 2017) penalizes the Kullback-Leibler divergence (KL divergence) using weighted hyper-parameters. Factor VAE (Kim and Mnih, 2018) and β-TCVAE (Chen et al., 2018) are trained using total correlation (TC) to encourage independent dimensions in a latent vector, employing a discriminator and decomposed components of the KL divergence term. However, these works are based on the standard Gaussian prior.

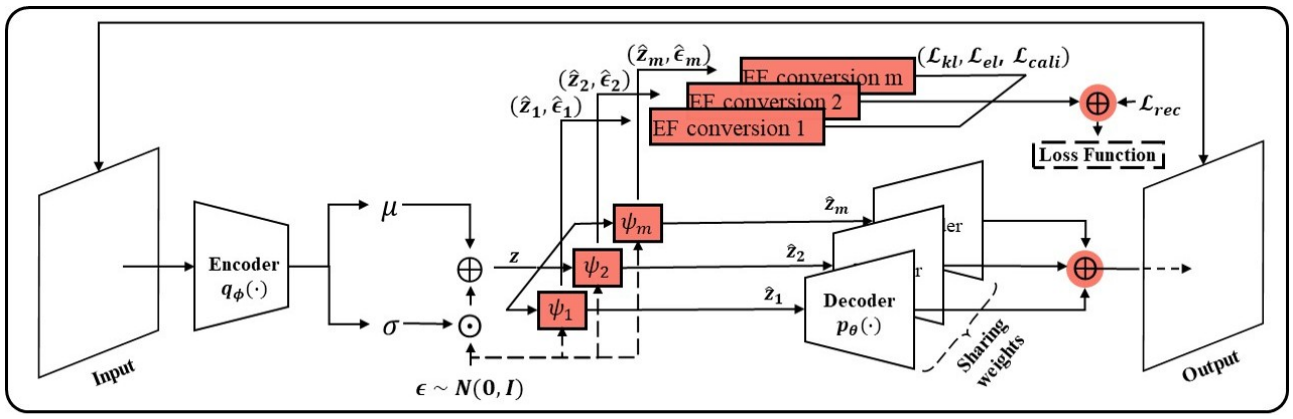


Figure 1: The overall architecture of our proposed *MIPET*-VAE. The invertible and partial-equivariant function $\psi(\cdot)$ for latent-to-latent (L2L) transformation consists of a symmetric matrix exponential to be 1) invertible and 2) partial-equivariant. Then 3) EF conversion module converges the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

Other VAE approaches implement other prior from Gaussian distribution to transformed Gaussian distribution, Gaussian mixture distribution (Dilokthanakul et al., 2016) or von Mises-Fisher distribution (Davidson et al., 2018). Mathieu et al. (2018) shows that model regularization with a Gaussian prior is inappropriate to convey sufficient dataset information and for disentanglement learning. Rather than using a Gaussian distribution, we employed an exponential family with high flexibility that encompasses a range of different distributions. InteL-VAE (Miao et al., 2022) utilize the invertible function to transform a Gaussian distribution to a complex distribution as Mathieu et al. (2018). Differently, we show a clearer relation of invertibility to disentanglement and improve VAEs to use its unrestricted form of prior. Differently, we consider the recent disentanglement definition based on group theory (Higgins et al., 2018).

3.2 Inductive Bias with Group Theory-Based Approaches

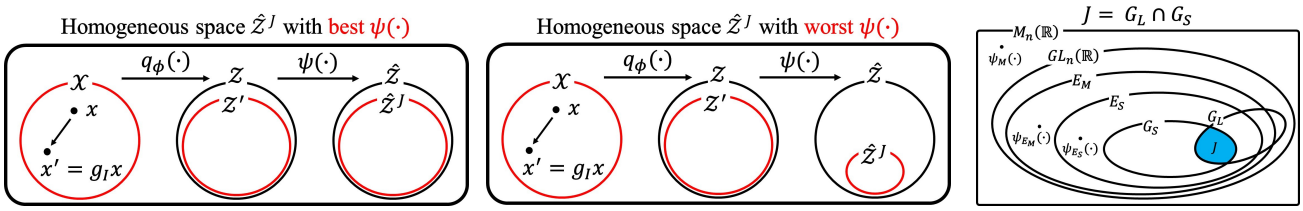
Following the definitions of disentangled representation learning based on group theory, several works have emphasized equivariant and improved disentangled representation learning. Commutative Lie Group VAE (CLG-VAE) (Zhu et al., 2021) proposed a direct mapping of the latent vector into Lie algebra to obtain a group structure (inductive bias) with constraints: commutative and hessian loss. Furthermore, Groupified VAE (Yang et al., 2022) utilizes the Spatial Broadcast Decoder (Watters et al., 2019) to implement an equivariant function to the cyclic group while guaranteeing the commutativity and invertibility of group actions. Topographic VAE (Keller and Welling, 2021) combines Student's-t distributions with variational inference and enforces rotated latent vectors to be equivariant. On the other hand, we apply an unrestricted prior and posterior for disentanglement learning.

4 Method

The overview of a VAE equipped with MIPE-transformation is shown in Fig. 1. We aim to explain the three core components of the MIPE-transformation. In Section 4.1, we mathematically prove why the *IPE-transformation*, which transforms latent vectors, better preserves partial-equivariance when it is an invertible and symmetric matrix. In Section 4.2, we describe *EF-conversion*, which extends the Gaussian distribution to a diverse exponential family distribution. Finally, in Section 4.3, we illustrate how multiple IPE-transformations and EF-conversions can be integrated into a VAE-based model for disentangled representation.

Notation We denote $g_I \in G_I, g_L \in G_L$, and $g_T \in G_T$. $M_n(\mathbb{R})$ is $n \times n$ real matrix, E_M is $\{\mathbf{e}^M | M \in M_n(\mathbb{R})\}$, E_S is a set of $\{\mathbf{e}^S | S \in Sym_n(\mathbb{R})\}$, and G_S is a group $G_S : (\mathbf{e}^S, *)$, where $*$ is a matrix multiplication (equal to commutative). Further details are in Appendix A.

Assumption Let's assume that 1) encoder $q_\phi(\cdot)$ is a partial-equivariant function, 2) $\mathcal{X}, \mathcal{Z}'$, and $\hat{\mathcal{Z}}'$ are homogeneous space of group G_I, G_L, and G_T, 3) $G_L \subset GL_n(\mathbb{R})$, where $GL_n(\mathbb{R})$ is the general linear group, and 4) matrices are restricted to a compact subset $K \subset M_n(\mathbb{R})$ and probabilities are defined with respect to the normalized uniform probability measure μ on K (a modeling assumption motivated by finite-precision implementations).



(a) Visualization of homogeneous space (red) as best and worst L2L transformation $\psi(\cdot)$ (b) Venn diagram of groups

Figure 2: The homogeneous space $\mathcal{Z}'$ is induced by the encoder q_ϕ, and the cardinality of the $\widehat{\mathcal{Z}}^J$ depends on the latent-to-latent (L2L) transformation.

4.1 Partial-Equivariant Function and Invertible L2L Transformation

Why Should L2L Transformation be Equivariant? We transform the latent vector $\boldsymbol{z}$ sampled from a Gaussian distribution to the transformed latent vector $\hat{\boldsymbol{z}}$ to be sampled from a different distribution. However, not all L2L transformations preserve the partial-equivariance of encoder condition. As illustrated in Fig. 2a, we choose ψ to *maximize* the cardinality of the homogeneous G_T^J-space induced in latent space:

$$\psi^\star = \arg\max_{\psi \in M_n(\mathbb{R})} \left|\widehat{\mathcal{Z}}^J(\psi)\right|, \tag{7}$$

where G_T^J act on the latent space via α and define $\widehat{\mathcal{Z}}^J(\psi) := \{\alpha(g, \hat{\boldsymbol{z}}) \mid g \in G_T^J\}$. As a homogeneous G_T^J–space, fixing a base point $\hat{\boldsymbol{z}}_0$ determines all elements $\alpha(g, \hat{\boldsymbol{z}}_0)$. If the latent space factors as $\hat{\mathcal{Z}} = \hat{\mathcal{Z}}_1 \times \cdots \times \hat{\mathcal{Z}}_k$ and the action is componentwise, then every $\alpha(g, \hat{\boldsymbol{z}}_0)$ with $g \in G_1 \times \cdots \times G_k$ is also disentangled (Higgins et al., 2018).

Equivariance Property with Symmetric Matrix Exponential We prove that an invertible and symmetric matrix preserves partial-equivariance better than other matrices for Eq. 7. We show that 1) a group with symmetric and invertible matrices guarantees equivariance of $\psi(\cdot)$ over the specific group G_S, 2) this $\psi(\cdot)$ being equivariant over subset of symmetries between the input space and transformed latent vector space (equal to be a partial-equivariant), and 3) the invertible and symmetric matrix (symmetric matrix exponential) increases the probability of $\psi(\cdot)$ to be in the group (equal to satisfy Eq. 7).

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

Proof. The group G_S is closed to matrix multiplication, and its element is always a symmetric matrix by definition. Then, any two elements in G_S are commutative because if matrix multiplication of two symmetric matrices is symmetric then both are commutative by Lemma B.1. 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 Eq. 1 by matching f, g, and G to ψ_{G_S}, g_s, and G_S, respectively. ■

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

Proof. The function $\psi_{G_S}(\cdot)$ is an equivariant function over group elements in $G_S \cap G_L$ by Proposition 4.1. Then, the composite function, $\psi_{G_S}(\cdot)$ and q_ϕ, is an equivariant function of G_I^J and G_T^J. Let $g_L^J \in G_L^J$, and $g_T^J \in G_T^J$. Then, $\psi_{G_S}(g_L^J \boldsymbol{z}_1) = g_L^J \psi_{G_S}(\boldsymbol{z}_1) = g_T^J \psi_{G_S}(\boldsymbol{z}_1)$. ■

Therefore, the invertible and symmetric matrix is a partial-equivariant function between the data space and transformed latent vector space. More details are in Proposition B.2.

Let $P(G) := \frac{\mu(G \cap K)}{\mu(K)}$ by the assumption 4 in section 4, and $Pr(\psi_{G'}(\cdot) \in G) = P(\psi(\cdot) \in G | \psi(\cdot) \in G')$ after VAE training.

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

Proof. All $\mathbf{e}^{\boldsymbol{S}} \in E_S$ are in E_M since $Sym_n(\mathbb{R}) \subset M_n(\mathbb{R})$. However, $E_M \not\subset E_S$ because $\mathbf{e}^{\boldsymbol{S}}$ is always symmetric, but $\mathbf{e}^{\boldsymbol{M}}$ can be an asymmetric matrix. Therefore $E_M \not\subset E_S$. Therefore, the conditional probability $Pr(\psi_{E_S}(\cdot) \in G_S) = \frac{P(G_S)}{P(E_S)}$ is greater than $Pr(\psi_{E_M}(\cdot) \in G_S) = \frac{P(G_S)}{P(E_M)}$. In the same way, $Pr(\psi_{E_M}(\cdot) \in G_S) > Pr(\psi_M(\cdot) \in G_S) = \frac{P(G_S)}{P(M_n(\mathbb{R}))}$ because $E_M \subset M_n(\mathbb{R})$ and non-invertible functions are only in $M_n(\mathbb{R})$. More details are in Proposition B.3. ■

Therefore, ψ_{E_S} clearly increases the probability of Eq. 7, because the group G_L is obtained ($|\hat{\mathcal{Z}}^J(\psi)|$ is fixed) by our assumption that $q_\phi(\cdot)$ is a partial-equivariant. It implies that symmetric and invertible properties for a matrix are necessary to preserve partial equivariance.

The probability $Pr(\psi_{E_S}(\cdot) \in G_S)$, $Pr(\psi_{E_M}(\cdot) \in G_S)$, and $Pr(\psi_M(\cdot) \in G_S)$ is changed by the distribution of the observation of $\psi(\cdot)$, which depends on the model parameters. However, the inequality $Pr(\psi_{E_S}(\cdot) \in G_S) > Pr(\psi_{E_M}(\cdot) \in G_S) > Pr(\psi_M(\cdot) \in G_S)$ is not changed regardless of the distribution of observation of $\psi(\cdot)$. We empirically validate the impact of equivariance with the uncertain $P(\cdot)$ to disentanglement in Section D.5.

Invertible Property by Using Matrix Exponential To guarantee the invertible property of IPE-transformation, we use a function $\psi(\cdot) = \mathbf{e}^{\boldsymbol{M}} * \cdot$ for the transformation, where $\boldsymbol{M}$ is in $n \times n$ real number matrix set $M_n(\mathbb{R})$ (Xiao and Liu, 2020). The operator $*$ is matrix multiplication, and $\mathbf{e}^{\boldsymbol{M}} = \sum_k^\infty \frac{\boldsymbol{M}^k}{k!}$. Our motivation is to use the benefits of injecting explicit inductive bias for disentanglement (Locatello et al., 2019; Miao et al., 2022). InteL-VAE effectively extracts hierarchical representation, which includes low-level features (affect to a specific factor) and high-level features (affect to complex factors) with an invertible transformation function (Miao et al., 2022).

4.2 Exponential Family Conversion for Unknown Prior

In VAE frameworks, the Gaussian normal distribution is applied as a prior. However, a prior from data is usually unknown and may not follow the Gaussian distribution (Miao et al., 2022). As a solution, we present a training procedure for VAEs to build an exponential family distribution from a latent variable of an arbitrary distribution. Then, we introduce training losses obtained from the unit IPE-transformation function and EF-conversion.

Elements of Exponential Family Distribution Settings As introduced in Bishop (2006); Charpentier et al. (2022), we assume exponential family always admits a conjugate prior Eq. 5. 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., 2022). 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 we follow Eq. 4.

Distribution Approximation As an Exponential Family By our exponential family settings, a posterior distribution in the exponential family by adopting the following Eq. 6. The *evidence* is implemented as learnable parameters $\boldsymbol{\nu} \in \mathbb{R}^{n \times n}$, and natural parameter is generated by a multi-layer perceptron as Charpentier et al. (2022). This general form approximating an exponential family distribution with learnable parameters can extend VAEs to use a wider distribution for latent variables by simply matching $\mathbf{X}$ to generated latent variables. After IPE-transformation, we can apply the form by using the $\hat{\boldsymbol{z}}_m$, $\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m}$, and $\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}_m}$ for $\mathbf{X}$, $\boldsymbol{\theta}$, and ξ, respectively as shown in Fig. 1.

EF Similarity Loss To maximize Eq. 6 to be an exponential family, we added a loss through a Lagrange multiplier as:

$$\text{maximize } \log p(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m}|\hat{\boldsymbol{z}}_m, \boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}_m}, \boldsymbol{\nu}_m) \quad \text{s.t. } D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m})||f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}_m})) \geq 0 \tag{8}$$

$$\begin{aligned} \Rightarrow \mathcal{L}_s(\hat{\boldsymbol{z}}_m, \hat{\boldsymbol{\epsilon}}_m) &= \log p(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m}|\hat{\boldsymbol{z}}_m, \boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}_m}, \boldsymbol{\nu}_m) \\ &\quad + \boldsymbol{\lambda}_m D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m})||f_{\mathbf{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}_m})) \end{aligned} \tag{9}$$

$$\Rightarrow \mathcal{L}_{el} \coloneqq ||\nabla_{\hat{\boldsymbol{z}}_m, \hat{\boldsymbol{\epsilon}}_m, \boldsymbol{\lambda}_m} \mathcal{L}_s||_2^2. \tag{10}$$

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.

KL Divergence for Evidence of Lower Bound We compute the KL divergence of the exponential family in Eq. 9 using the learnable parameter $T(\cdot)$ and $A(\cdot)$ with given natural parameter $\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}$ and $\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}$, expressed as:

$$\begin{aligned} \mathcal{L}_{kl} &\coloneqq D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m})||f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}_m})) \\ &= A(\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}) - A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) + \boldsymbol{\theta}_{\hat{\boldsymbol{z}}}^\intercal \nabla_{\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}} A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) - \boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}^\intercal \nabla_{\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}} A(\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}). \end{aligned} \tag{11}$$

Further details are in the Appendix B.1.

KL Divergence Calibration Loss Although KL is invariant under a common invertible map ψ (Miao et al., 2022), practical approximations (e.g., matrix-exponential) can break this. We therefore add a consistency regularizer that penalizes the gap between empirical KLs before and after applying ψ:

$$\mathcal{L}_{\text{cali}} = \mathrm{MSE}(D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \,||\, p_\theta(\boldsymbol{z})), D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m}) \,||\, f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}_m}))) \tag{12}$$

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

Implicit Semantic Mask We propose an implicit semantic mask to improve disentanglement learning.

Table 1: Performance (mean ± std) of four metrics on dSprites, 3D Shapes, and 3D Cars.

dSprites	FVM ↑		MIG ↑		SAP ↑		DCI ↑	
	original	MIPET	original	MIPET	original	MIPET	original	MIPET
β-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)
β-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)
Factor-VAE	67.78(±7.48)	**68.38**(±8.55)	14.67(±10.40)	**19.31**(±13.46)	2.35(±2.32)	**4.24**(±3.36)	22.58(±8.50)	**30.18**(±12.99)
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)
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	FVM ↑		MIG ↑		SAP ↑		DCI ↑	
	original	MIPET	original	MIPET	original	MIPET	original	MIPET
β-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)
β-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)
Factor-VAE	77.43(±10.71)	**78.34**(±8.35)	24.23(±26.13)	**48.59**(±10.58)	4.99(±4.46)	**9.84**(±2.73)	**68.68**(±12.92)	60.23(±9.90)
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)
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	FVM ↑		MIG ↑		SAP ↑		DCI ↑	
	original	MIPET	original	MIPET	original	MIPET	original	MIPET
β-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)
β-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)
Factor-VAE	89.64(±2.62)	**93.66**(±3.83)	**10.63**(±1.30)	9.43(±1.12)	2.51(±0.37)	**3.28**(±0.83)	**26.27**(±3.17)	23.13(±0.72)
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)
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)

We apply mask matrix $\mathcal{M}$ which consists of 0 or 1 element to log-normalizer to prevent less effective weight flow as:

$$\mathcal{M}_{ij} = \begin{cases} 1 \text{ if } |\mathcal{W}_{ij}| \geq \mu_{|\mathcal{W}_{ij}|} - \lambda\sigma_{|\mathcal{W}_{ij}|} \\ 0 \text{ otherwise} \end{cases}, \quad (13)$$

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

4.3 Integration for Multiple IPE-Transformation and EF-Conversion

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

$$\mathcal{L}_{\text{rec}} = \frac{1}{k}\sum_{i=1}^{k} \mathbb{E}_{q_{\phi,\psi_i}(\boldsymbol{z}|\boldsymbol{x})} \log p_\theta(\boldsymbol{x}|\psi_i(\boldsymbol{z})) \quad (14)$$

where $\hat{\boldsymbol{z}}_i = \psi_i(\boldsymbol{z})$. Therefore, we define ELBO as:

$$\mathcal{L}'(\phi,\theta,\psi_{i\in[1,k]};\boldsymbol{x}) = \frac{1}{k}\sum_{i=1}^{k} \mathbb{E}_{q_{\phi,\psi_i}(\boldsymbol{z}_i|\boldsymbol{x})} \log p_\theta(\boldsymbol{x}|\psi_i(\boldsymbol{z})) - \sum_{i=1}^{k} D_{\text{KL}}(q_{\phi,\psi_i}(\boldsymbol{z}|\boldsymbol{x})||p_{\psi_i}(\boldsymbol{z})). \quad (15)$$

However, following Eq. 15, 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. 15 represents that increasing number of MIPE-transformation is equal to an increasing β hyper-parameter in β-VAE (Higgins et al., 2017) and more details are in Appendix C.

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

$$\mathcal{L}(\phi,\theta,\psi_{i\in[1,k]};\boldsymbol{x}) = \mathcal{L}_{rec} - \mathcal{L}_{kl} - \mathcal{L}_{el} - \mathcal{L}_{cali}. \quad (16)$$

5 Experiment Settings

5.1 Models

As baseline models, we select VAE (Kingma and Welling, 2013), β-VAE (Higgins et al., 2017), β-TCVAE (Chen et al., 2018), Factor-VAE (Kim and Mnih, 2018), Control-VAE (Shao et al., 2020),and CLG-VAE (Zhu et al., 2021). These models are compared to their extension to adopt MIPET, abbreviated by adding the MIPET prefix. Also, we follow the same model architecture which are introduced in previous works (Kim and Mnih, 2018) and model details are in Table 6-7.

5.2 Datasets

As in prior works (Chen et al., 2018; Higgins et al., 2017; Keller and Welling, 2021; Kim and Mnih, 2018; Yang et al., 2022; Zhu et al., 2021), we primarily evaluate our models on synthetic datasets, where

Table 2: Impact of the number of MIPE-transformation function on the β-TCVAE and β-VAE with dSprites, 3D Shapes, and 3D Cars datasets in terms of the four metrics. The blue and red box plots represent each model's single and multiple IPE-transformation cases, respectively (A-n: MIPET-β-TCVAE (4), B-n: MIPET-β-TCVAE (6), C-n: MIPET-β-VAE, n: the number of MIPE-transformation). Further details are in Table 9.

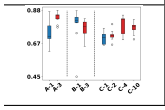
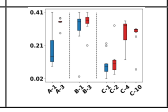
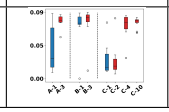
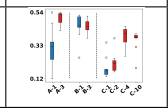
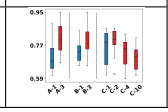
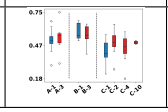
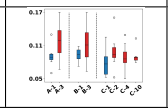
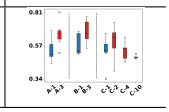
dSprites				3D Shapes			
FVM	MIG	SAP	DCI	FVM	MIG	SAP	DCI
							

Table 3: Impact of the mask (mean±std.) and its ratio λ in Eq. 13 on 3D Cars. (∞: no masking case, gray box: the best setting over all metrics, bold text: the best in each metric.) Each model runs with ten random seeds.

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

learning disentangled representations remains a well-known challenge (Burgess and Kim, 2018; Matthey et al., 2017; Reed et al., 2015). We benchmark MIPET-VAEs against several standard VAEs on the dSprites (Matthey et al., 2017), 3D Shapes (Burgess and Kim, 2018), and 3D Cars (Reed et al., 2015) datasets. Further details are in the Appendix D.2

5.3 Training

We set 256 mini-batch size in the datasets (dSprites, 3D Shapes, and 3D Cars), Adam optimizer with learning rate 4×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and epochs from $\{30, 67, 200\}$ as a common setting for all the comparative methods. In addition, we set masking ratio λ from $\{0.0, 0.5, \ldots 2.0, \infty\}$. We then evaluate familiar benchmarks, FVM score (Kim and Mnih, 2018), MIG (Chen et al., 2018), SAP (Kumar et al., 2018), and DCI (Eastwood and Williams, 2018). Further details are in the Appendix D.3.

6 Results and Discussion

6.1 Quantitative Analysis

Disentanglement Metrics We set the number of IPE-transformation functions to be equal to balancing hyper-parameter β on β-VAE because of Eq. 16. The number of IPE-transform functions of β-TCVAE is 3. However, in the case of CLG-VAE, we set it to 1 because its approach is based on the group theory, not directly controlling a KL divergence term such as β-VAE. We average each model performance value with 40, 20, 60, 10, and 30 cases in VAEs, β-TCVAEs, Control-VAE, Factor-VAE and CLG-VAEs, respectively.

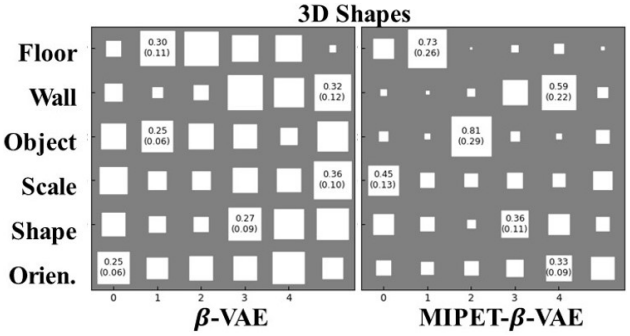


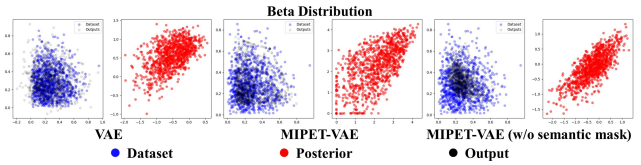
Figure 3: Each square represents a value in the DCI matrix, which describes the relationship between the i^{th} latent dimension and each factor. The size of each square is relative to the values within each row. The ideal case resembles a sparse matrix. The y-axis represents the factors of each dataset, while the x-axis corresponds to the latent vector dimensions. The number shown in each row of the matrix indicates the maximum value and standard deviation of that row. Higher maximum and standard deviation values suggest greater sparsity, indicating closer alignment with the ideal case.

As shown in Table 1, MIPET-VAEs disentanglement performance is broadly improved with four metrics on each dataset. In particular, most FVM results significantly affect the model performance and stability on all datasets. Therefore, our proposed method obtains a specific dimension that corresponds to a specific single factor. These results imply that applied to MIPE-transformation functions on VAEs elaborate disentangled representation learning.

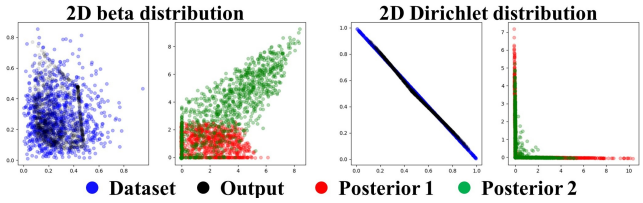
Sensitivity to the Number of IPE-transformation and EF-conversion We analyze

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 β-VAE and β-TCVAE, respectively.

3D Shapes	β-VAE			β-TCVAE			Factor-VAE			CLG-VAE		
	MIPET	w/o E	w/o EF	MIPET	w/o E	w/o EF	MIPET	w/o E	w/o EF	MIPET	w/o E	w/o EF
FVM	**75.19**(±8.16)	74.91(±10.46)	22.27(±1.29)	**80.59**(±8.57)	77.90(±8.66)	66.38(±7.57)	**78.34**(±8.35)	71.50(±6.78)	59.28(±5.15)	**80.17**(±8.43)	69.75(±6.91)	61.55(±9.05)
MIG	47.37(±10.13)	**47.45**(±8.98)	0.28(±0.09)	**54.49**(±9.44)	51.37(±11.54)	36.08(±17.42)	**48.59**(±8.35)	42.25(±11.28)	32.67(±10.49)	**53.87**(±7.41)	45.92(±4.77)	33.89(±3.46)
SAP	9.20(±2.44)	**9.43**(±2.59)	0.26(±0.07)	**11.58**(±3.32)	10.23(±3.13)	7.13(±3.09)	**9.84**(±2.73)	8.19(±0.69)	7.04(±0.87)	**12.83**(±3.01)	50.14(±5.43)	46.66(±4.81)
DCI	**54.95**(±8.99)	54.23(±9.05)	0.10(±0.02)	**66.22**(±7.32)	61.18(±8.87)	56.85(±11.72)	**60.23**(±9.90)	50.65(±6.85)	48.83(±5.37)	**60.74**(±7.77)	50.14(±5.43)	46.66(±4.81)



(a) Posterior visualization with Beta distribution



(b) Posterior visualization of MIPET layer

Figure 4: Non-Gaussian posterior learned by an IPE module without intended guidance to a specific distribution in a toy setting. We compare VAE, MIPET-VAE, and MIPET-VAE without a semantic mask to assess how well each model captures the underlying distribution. We construct the VAE with a 4-layer Multi-Layer Perceptron (MLP) as the encoder and a single linear layer as the decoder. Blue plots are randomly sampled from a two-dimensional beta distribution, red plots are the posterior, and black plots are the output results.

the impact of the MIPE-transformation function. As presented in Table 2, MIPE-transformation outperforms IPE-transformation in disentanglement learning across all datasets. Indeed, the results of MIPET-β-VAEs more generally and clearly demonstrate the impact of the MIPE-transfomation function. Our derivation in Section C clearly explains the impact of MIPE-transformation. This result shows the impact of the multiple uses of IPE-transformation and EF-conversion.

Impact of Implicit Semantic Mask We set masking hyper-parameter λ from $\{0.0, 0.5, \cdots, 2.0, \infty\}$, and each model has different λ for best case. In Table 3, VAE and CLG-VAE with masked log-normalizer show better and well-balanced results than the models without masking, which implies improvement of disentanglement.

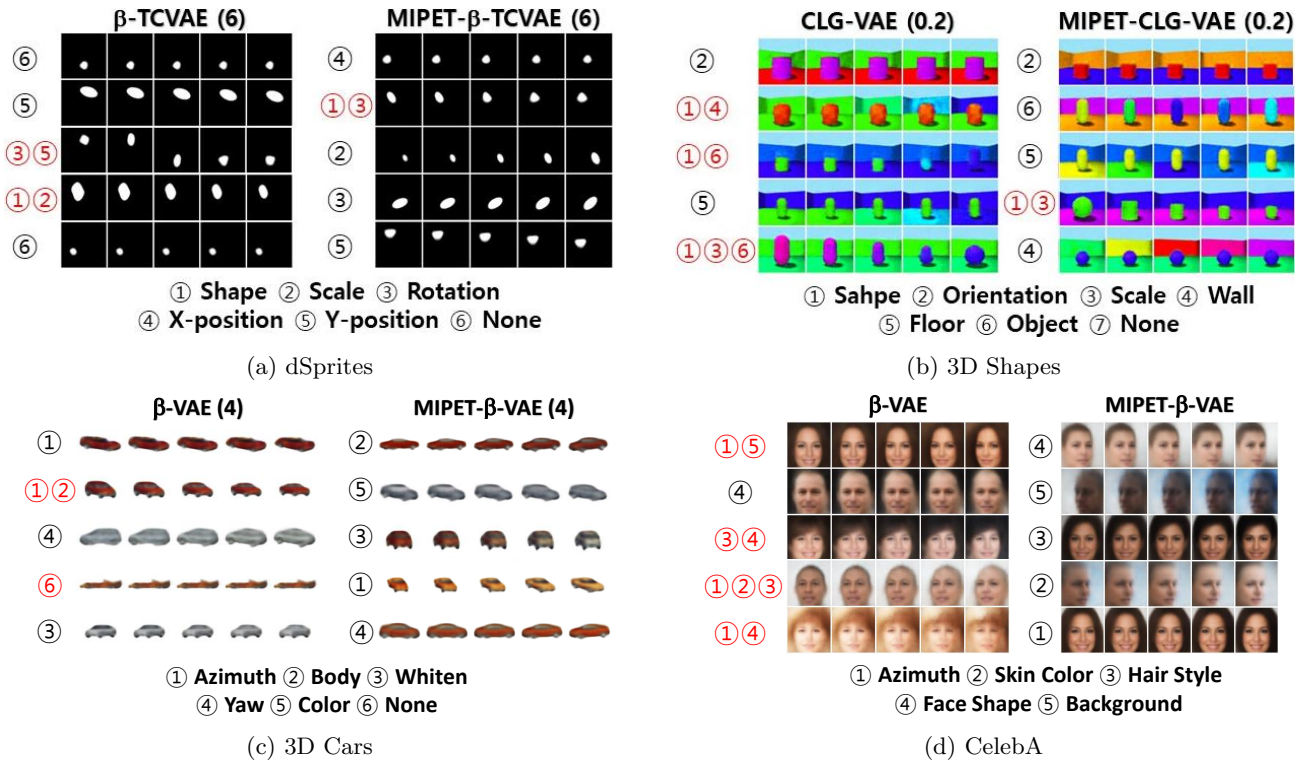
Ablation Study We conduct an ablation study to evaluate the separate impact of equivariant property and the EF-conversion. We have already presented the impact of the multiple uses of IPE-transform and EF-conversion in the previous paragraph. We evaluate the impact of the other properties by setting MIPE-transformation 1) without equivariant (w/o E), which is implemented as an asymmetric matrix, and 2) without EF-conversion (w/o EF). To exclude group theory interference with other methods, we select β-VAE and β-TCVAE. As the results are shown in Table 4, most of the results show that MIPET-VAEs performance is better than other cases. In particular, MIPET (w/o EF) results are lower than MIPET (w/o E) results and are clearly shown in all cases.

6.2 Qualitative Analysis

Relation Between Factors and Latent Vector Dimensions As shown in Fig. 3, our model shows a close to sparse matrix compared to the β-VAE result in 3D Shapes, also the maximum value and standard deviation of each row are larger than β-VAE. This indicates that our model achieves better alignment of individual factors with specific latent dimensions compared to other methods. More details are in Appendix D.4.

The Impact of IPE-transformation The IPE-transformation $\psi(\cdot)$ is designed to achieve two objectives: 1) to represent a non-Gaussian exponential family distribution as the posterior, and 2) to preserve the inherent structure of the dataset. To assess the first objective, we construct a toy dataset sampled from a beta distribution (additional cases are provided in Appendix D.7) and configure the decoder as a single linear layer, noting that the linear transformation of a Beta distribution remains within the same family. As shown in Fig. 4a, the posterior learned by MIPET-VAE (red dots) closely approximates the beta distribution, whereas the posteriors of VAE and MIPET (without the semantic mask) retain Gaussian distribution. Regarding the second objective, MIPET-VAE exhibits improved preservation of the dataset structure relative to VAE (black dots). Additional experimental details are reported in Appendix D.7.

Does Each IPE-transformation Independently Operate? We demonstrate how each IPE-transformation represents the exponential family. As shown in Fig. 4b, each IPE-transformation represents a different distribution because the green and red plots are distributed in different posteriors,



(a) dSprites (b) 3D Shapes

(c) 3D Cars (d) CelebA

Figure 5: Qualitative results on various datasets, which show the factors learned for each dimension of z.

even though all IPE-transformations share the same decoder weights.

Disentanglement of Factors Across Latent Dimensions We randomly sample an image for each dimension of the latent vector space and creates 5 variants of its generated latent vector by selecting values from {-2, 2} with 5 intervals for the dimension, then generate their corresponding output images. For the generation, we select β-TCVAE (6), which shows the best FVM scores in dSprites dataset. Thereafter, we evaluate the semantic roles of each dimension before and after applying MIPE-transformation function.

In Fig. 5a, β-TCVAE struggles with rotation and y-position, as shown on the 3^{rd} row, and with scale and shape represented on the 4^{th} row. On the contrary, MIPET-β-TCVAE separates rotation and y-position factor (4^{th}, and 5^{th} rows). Applied our method on β-TCVAE shows better disentangled representation on dSprites dataset. These results also show that our proposed method improves disentangled representation learning. As shown in the Fig. 5b, CLG-VAE struggles with shape and wall color factors in 2^{nd} dimension, and shape and object color factors in 3^{rd} dimension. In particular, it struggles with three factors in 5^{th} dimension. On the other hand, MIPET-CLG-VAE separates shape, wall, and object color factors. The qualitative analysis with 3D Cars dataset, as shown in Fig. 5c, the left side is the β-TCVAE result, and it struggles with azimuth and body factors shown in the 2^{nd} row. However, MIPET-β-TCVAE separates azimuth (4^{th} row) and body (1^{st} row). In particular, MIPET-β-TCVAE learns *color* factor (2^{rd} row) which does not exist on β-TCVAE. Furthermore, our model shows improvement on the CelebA Liu et al. (2015) dataset, as shown in Fig. 5d. β-VAE struggles with most factors of the dataset, but ours separates all factors.

7 Conclusion

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

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.2022R1A2C2012054, Development of AI for Canonicalized Expression of Trained Hypotheses by Resolving Ambiguity in Various Relation Levels of Representation Learning).

References

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828.

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

Burgess, C. and Kim, H. (2018). 3d shapes dataset. https://github.com/deepmind/3dshapes-dataset/.

Charpentier, B., Borchert, O., Zügner, D., Geisler, S., and Günnemann, S. (2022). Natural posterior network: Deep bayesian predictive uncertainty for exponential family distributions. In *International Conference on Learning Representations*.

Chen, R. T. Q., Li, X., Grosse, R. B., and Duvenaud, D. K. (2018). Isolating sources of disentanglement in variational autoencoders. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

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

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

Eastwood, C. and Williams, C. K. I. (2018). A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*.

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

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

Jeong, Y. and Song, H. O. (2019). Learning discrete and continuous factors of data via alternating disentanglement. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3091–3099. PMLR.

Jung, H.-J., Jeong, J., and Kim, K. (2024). CFASL: Composite factor-aligned symmetry learning for disentanglement in variational autoencoder. *Transactions on Machine Learning Research*.

Keller, T. A. and Welling, M. (2021). Topographic vaes learn equivariant capsules. *CoRR*, abs/2109.01394.

Keurti, H., Pan, H.-R., Besserve, M., Grewe, B. F., and Schölkopf, B. (2023). Homomorphism AutoEncoder – learning group structured representations from observed transitions. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 16190–16215. PMLR.

Kim, H. and Mnih, A. (2018). Disentangling by factorising. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2649–2658. PMLR.

Kim, M., Wang, Y., Sahu, P., and Pavlovic, V. (2019). Bayes-factor-vae: Hierarchical bayesian deep auto-encoder models for factor disentanglement. In *IEEE International Conference on Computer Vision*.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes.

Kumar, A., Sattigeri, P., and Balakrishnan, A. (2018). VARIATIONAL INFERENCE OF DISENTANGLED LATENT CONCEPTS FROM UNLABELED OBSERVATIONS. In *International Conference on Learning Representations*.

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

Li, Z., Murkute, J. V., Gyawali, P. K., and Wang, L. (2020). Progressive learning and disentanglement of hierarchical representations. In *International Conference on Learning Representations*.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738.

Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. (2019). Challenging common assumptions in the unsupervised learning of disentangled representations. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4114–4124. PMLR.

Locatello, F., Poole, B., Rätsch, G., Scholkopf, B., Bachem, O., and Tschannen, M. (2020). Weakly-supervised disentanglement without compromises. In *International Conference on Machine Learning*.

Mathieu, E., Rainforth, T., Siddharth, N., and Teh, Y. (2018). Disentangling disentanglement in variational autoencoders. In *International Conference on Machine Learning*.

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

Miao, N., Mathieu, E., N, S., Teh, Y. W., and Rainforth, T. (2022). On incorporating inductive biases into VAEs. In *International Conference on Learning Representations*.

Michlo, N. J. (2021). Disent - a modular disentangled representation learning framework for pytorch. Github.

Reed, S. E., Zhang, Y., Zhang, Y., and Lee, H. (2015). Deep visual analogy-making. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Ren, X., Yang, T., Wang, Y., and Zeng, W. (2022). Learning disentangled representation by exploiting pretrained generative models: A contrastive learning view. In *ICLR*.

Shao, H., Yao, S., Sun, D., Zhang, A., Liu, S., Liu, D., Wang, J., and Abdelzaher, T. (2020). ControlVAE: Controllable variational autoencoder. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8655–8664. PMLR.

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

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

Togo, R., Nakagawa, N., Ogawa, T., and Haseyama, M. (2025). Concvae: Conceptual representation learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(4):7529–7541.

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

Winter, R., Bertolini, M., Le, T., Noe, F., and Clevert, D.-A. (2022). Unsupervised learning of group invariant and equivariant representations. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.

Xiao, C. and Liu, L. (2020). Generative flows with matrix exponential. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10452–10461. PMLR.

Yang, T., Ren, X., Wang, Y., Zeng, W., and Zheng, N. (2022). Towards building a group-based unsupervised representation disentanglement framework. In *International Conference on Learning Representations*.

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

Zhu, X., Xu, C., and Tao, D. (2021). Commutative lie group vae for disentanglement learning. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12924–12934. PMLR.

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

Checklist

The checklist follows the references. For each question, choose your answer from the three possible options: Yes, No, Not Applicable. You are encouraged to include a justification to your answer, either by referencing the appropriate section of your paper or providing a brief inline description (1-2 sentences). Please do not modify the questions. Note that the Checklist section does not count towards the page limit. Not including the checklist in the first submission won't result in

desk rejection, although in such case we will ask you to upload it during the author response period and include it in camera ready (if accepted).

In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes - We present in Section 4, and Appendix B.]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes - We present in the Appendix D.6.]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes - We submit our code as supplement materials.]
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes - We present in Section 4, and Appendix B.]
   (b) Complete proofs of all theoretical results. [Yes - We present in Section 4, and Appendix B.]
   (c) Clear explanations of any assumptions. [Yes - We present in Section 4, and Appendix B.]
3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes - We submit our code as supplement materials.]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes - We present in Section 5 and Appendix D.]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes - We present in Section 6.]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes - We present in Appendix D.]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes - We submit our code as supplement materials.]
   (b) The license information of the assets, if applicable. [Yes - We submit our code as supplement materials.]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes - We submit our code as supplement materials.]
   (d) Information about consent from data providers/curators. [Yes - We present in Appendix D.]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

MIPET Supplementary Materials

A Notation and Groups

Table 5: Terms and Notations

$\psi(\cdot)$	Invertible function
$\boldsymbol{z}$	Latent vector from encoder
$\hat{\boldsymbol{z}}_m$	Transformed latent vector by $\psi_m(\cdot)$
$\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m}$	Natural Parameter of posterior
$\hat{\boldsymbol{\epsilon}}_m$	Transformed prior samples by $\psi_m(\cdot)$
$\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}_m}$	Natural Parameter of prior
A	Log-Normalizer
T	Sufficient Statistics
$\boldsymbol{\nu}$	Evidence
$f_{\boldsymbol{x}}(\cdot)$	Power Density Function
$GL_n(\mathbb{R})$	General Linear Group
$M_n(\mathbb{R})$	A set of $n \times n$ real matrix
$Sym_n(\mathbb{R})$	A set of $n \times n$ symmetric real matrix
E_M	$\{\mathbf{e}^M \mid M \in M_n(\mathbb{R})\}$
E_S	$\{\mathbf{e}^S \mid S \in Sym_n(\mathbb{R})\}$
$\psi_M(\cdot)$	$\psi_M(\cdot) \in M_n(\mathbb{R})$
$\psi_{E_S}(\cdot)$	$\psi_{E_S}(\cdot) \in E_S$
$\psi_{E_M}(\cdot)$	$\psi_{E_M}(\cdot) \in E_M$
G_S	$G_S : (\mathbf{e}^S, *)$
G_I	Group of input space for symmetries
G_L	Group of latent space for symmetries
G_T	Group of transformed latent space for symmetries
J	$G_S \cap G_L$
G_I^J	Group of input space for symmetries corresponds to J
G_L^J	Group of latent space for symmetries corresponds to J
G_L^J	Group of transformed latent space for symmetries corresponds to J
$\mathcal{X}$	Input space
$\mathcal{Z}$	Latent vector space
$\hat{\mathcal{Z}}$	Transformed latent vector space
$\mathcal{Z}'$	Homogeneous space of group G_L (homogeneous G_L-space), $\mathcal{Z}' \subseteq \mathcal{Z}$
$\hat{\mathcal{Z}}'$	Homogeneous space of group G_T (homogeneous G_T-space), $\hat{\mathcal{Z}}' \subseteq \hat{\mathcal{Z}}$
$\hat{\mathcal{Z}}^J$	Homogeneous space of group G_T^J (homogeneous G_T^J-space), $\hat{\mathcal{Z}}^J \subseteq \hat{\mathcal{Z}}$
$D_{\mathrm{KL}}(\cdot\|\|\cdot)$	Kullback-Leibler divergence
$\mathbf{0}$	zero vector
$\mathbf{0}_{n,n}$	n by n zero matrix

Notation. We denote $g_I \in G_I$, $g_L \in G_L$, and $g_T \in G_T$. Let $M_n(\mathbb{R})$ be the set of $n \times n$ real matrices and $\mathrm{Sym}_n(\mathbb{R})$ the set of real symmetric matrices. For any set $\mathcal{A} \subseteq \mathrm{Sym}_n(\mathbb{R})$, define $\exp(\mathcal{A}) := \{\exp(S) \mid S \in \mathcal{A}\}$. We write $E_M := \{\exp(M) \mid M \in M_n(\mathbb{R})\}$ and $E_S := \exp(\mathrm{Sym}_n(\mathbb{R}))$.

Standing assumption on G_S. We assume a *pairwise-commuting* symmetric set $\mathcal{S} \subseteq \mathrm{Sym}_n(\mathbb{R})$, i.e., $[S_1, S_2] = 0$

for all $S_1, S_2 \in \mathcal{S}$. Then we define

$$G_S := \exp(\mathcal{S}) = \{\exp(S) \mid S \in \mathcal{S}\} \subset \mathrm{GL}_n(\mathbb{R}),$$

with the group operation given by the usual matrix multiplication "$*$".

B Proof

Lemma B.1 (Abelian Lie subgroup)**.** *If $\mathcal{S} \subseteq \mathrm{Sym}_n(\mathbb{R})$ is pairwise-commuting, then $(G_S, *)$ is an abelian Lie subgroup of $\mathrm{GL}_n(\mathbb{R})$. In particular, G_S is closed under multiplication and inversion, has identity $I = \exp(0)$, and is commutative.*

Proof sketch. Since all $S \in \mathcal{S}$ commute and are symmetric, they are simultaneously diagonalizable by some orthogonal matrix Q: $S_i = Q\Lambda_i Q^\top$ with diagonal Λ_i. Hence $\exp(S_i) = Q\exp(\Lambda_i)Q^\top$ are symmetric positive definite and commute. Moreover, for any $S_1, S_2 \in \mathcal{S}$, the Baker–Campbell–Hausdorff formula reduces to $\exp(S_1)\exp(S_2) = \exp(S_1 + S_2)$ because $[S_1, S_2] = 0$. Since $S_1 + S_2 \in \mathcal{S}$, we have closure. The inverse is $\exp(S)^{-1} = \exp(-S)$ and $-S \in \mathcal{S}$. Commutativity follows from $[S_1, S_2] = 0$. Smoothness of exp implies G_S is a (abelian) Lie subgroup. □

Proposition B.2. *By the Lemma B.1, if q_ϕ is equivariant over defined on group of symmetries G_I^J and G_L^J, then $\psi_{G_S}(q_\phi(\cdot))$ is equivariant to symmetries in G_I corresponding to $G_S \cap G_L$ and G_T corresponding to $G_S \cap G_L$ by the equivariance of q_ϕ.*

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

$$\hat{\boldsymbol{z}}_1 = \psi_{G_S}(\boldsymbol{z}_1), \text{and} \tag{17}$$

$$\hat{\boldsymbol{z}}_2 = \psi_{G_S}(\boldsymbol{z}_2) = \psi_{G_S}(g_L^J \boldsymbol{z}_1) = g_L^J \psi_{G_S}(\boldsymbol{z}_1) \ (\because \text{Prop. 4.1}), \tag{18}$$

$$\begin{aligned} &\text{then } g_L^J \psi_{G_S}(\boldsymbol{z}_1) = g_T^J \psi_{G_S}(\boldsymbol{z}_1) \ (\because \hat{\boldsymbol{z}}_2 = g_T^J \hat{\boldsymbol{z}}_1) \\ &\Rightarrow (g_L^J - g_T^J)\psi_{G_S}(\boldsymbol{z}_1) = \mathbf{0}, \end{aligned} \tag{19}$$

where $\mathbf{0}$ is a zero vector. Eq. 19 is defined when $\forall \boldsymbol{z} \in \mathcal{Z}$ by the equivariance definition. In other words, Eq. 19 is satisfied only if the kernel (linear algebra) of $g_L^J - g_T^J$, notated as $ker(g_L^J - g_T^J)$, includes the basis of $\mathbb{R}^n$ vector space. If the standard basis of $\mathbb{R}^n$ vector space is in $ker(g_L^J - g_T^J)$, then $(g_L^J - g_T^J) = \mathbf{0}_{n,n}$, 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 $g_L^J - g_T^J = \mathbf{0}_{n,n}$.

Then, $\psi_{G_S}(g_L^J \boldsymbol{z}_1) = g_L^J \psi_{G_S}(\boldsymbol{z}_1) = g_T^J \psi_{G_S}(\boldsymbol{z}_1)$. The encoder is an equivariant function over input space $\mathcal{X}$ as $q_\phi(g_I^J \boldsymbol{x}_1) = g_L^J q_\phi(\boldsymbol{x}_1)$. Mixing two equivarience property, we can derive another equivariance relation $g_T^J \psi_{G_S}(q_\phi(\boldsymbol{x}_1)) = \psi_{G_S}(q_\phi(g_I^J \boldsymbol{x}_1))$ This result implies that the equivariance between input space and a latent space is preserved for $G_S \cap G_L$ if the latent vector $\boldsymbol{z}$ is transformed by ψ_{G_S}.■

We show that ψ_{G_S} preserves equivariance between G_L^J and G_I^J. If there exists equivariant function between input and latent vector space, there should be a group G_L for a latent space and its corresponding group G_I in an input space by definition of equivariance ($q_\phi(g_I x) = g_L q_\phi(x)$). In other words, $\psi_{G_S}(\cdot)$ guarantees to preserve the equivariance of I2L-transformation to certain symmetries in $G_S \cap G_L$ after IPE-transformation as shown in Fig. 2.

Let $P(G) := \frac{\mu(G \cap K)}{\mu(K)}$ by the assumption 4 in section 4, and $Pr(\psi_{G'}(\cdot) \in G) = P(\psi(\cdot) \in G | \psi(\cdot) \in G') = \frac{P(G \cap G')}{P(G')}$ after VAE training. Then,

Proposition B.3. *Under Assumption 4, the following probability inequality holds: $Pr(\psi_{E_S}(\cdot) \in G_S) > Pr(\psi_{E_M}(\cdot) \in G_S) > Pr(\psi_M(\cdot) \in G_S)$, where probabilities are defined with respect to the normalized uniform measure on a compact subset $K \subset M_n(\mathbb{R})$.*

Proof. Since $G_S \subset E_S \subset E_M \subset M_n(\mathbb{R})$,

$$\Pr(\psi_{E_S}(\cdot) \in G_S) = \frac{P(G_S \cap E_S)}{P(E_S)} = \frac{P(G_S)}{P(E_S)} = \frac{\mu(G_S \cap K)}{\mu(E_S \cap K)}, \ \left(\because G_S \subset E_S\right)$$

$$\Pr(\psi_{E_M}(\cdot) \in G_S) = \frac{P(G_S \cap E_M)}{P(E_M)} = \frac{P(G_S)}{P(E_M)} = \frac{\mu(G_S \cap K)}{\mu(E_M \cap K)}, \ \left(\because G_S \subset E_M\right)$$

$$\Pr(\psi_M(\cdot) \in G_S) = \frac{P(G_S \cap M_n(\mathbb{R}))}{P(M_n(\mathbb{R}))} = \frac{P(G_S)}{P(M_n(\mathbb{R}))} = \frac{\mu(G_S \cap K)}{\mu(M_n(\mathbb{R}) \cap K)}, \ \left(\because G_S \subset M_n(\mathbb{R})\right)$$

where μ is the normalized uniform measure on K. Because $G_S \subseteq E_S \subseteq E_M \subseteq M_n(\mathbb{R})$ and all denominators are finite by compactness of K, the ratios satisfy

$$\frac{\mu(G_S \cap K)}{\mu(E_S \cap K)} > \frac{\mu(G_S \cap K)}{\mu(E_M \cap K)} > \frac{\mu(G_S \cap K)}{\mu(M_n(\mathbb{R}) \cap K)}.$$

Thus the stated inequality follows. ■

B.1 KL Divergence for Evidence of Lower Bound

The KL divergence of Gaussian distribution (Kingma and 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. 6 using the learnable parameter $T(\cdot)$ and $A(\cdot)$ with given natural parameter $\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}$ and $\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}$, expressed as:

$$\begin{aligned}\mathcal{L}_{kl} &\coloneqq D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}_m})||f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}_m})) \\ &= A(\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}) - A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) + \boldsymbol{\theta}_{\hat{\boldsymbol{z}}}^{\intercal}\nabla_{\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}}A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) - \boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}^{\intercal}\nabla_{\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}}A(\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}).\end{aligned} \tag{20}$$

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

$$\begin{aligned}D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})||f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})) &= \int_{-\infty}^{\infty} f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) \log f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\mathrm{d}\boldsymbol{x} \\ &\quad - \int_{-\infty}^{\infty} f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) \log f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})\mathrm{d}\boldsymbol{x}.\end{aligned} \tag{21}$$

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

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

Expanding the Eq. 22, and $\int_{-\infty}^{\infty} f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\mathrm{d}\boldsymbol{x} = 1$, then

$$\begin{aligned}\int_{-\infty}^{\infty} f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) \log f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\mathrm{d}\boldsymbol{x} &= -A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) + \boldsymbol{\theta}_{\hat{\boldsymbol{z}}}^{\intercal} \int_{-\infty}^{\infty} T(\boldsymbol{x}) f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\mathrm{d}\boldsymbol{x} \\ &\quad + \int_{-\infty}^{\infty} B(\boldsymbol{x}) f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\mathrm{d}\boldsymbol{x}.\end{aligned} \tag{23}$$

Expanding $\int_{-\infty}^{\infty} f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) \log f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})\mathrm{d}\boldsymbol{x}$ using Eq. 22 and 23 and plug into Eq. 21 with Eq. 23, we have

$$\begin{aligned}D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})||f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})) &= A(\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}) - A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) \\ &\quad + \boldsymbol{\theta}_{\hat{\boldsymbol{z}}} \int_{-\infty}^{\infty} T(\boldsymbol{x}) f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})\mathrm{d}\boldsymbol{x} \\ &\quad - \boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}} \int_{-\infty}^{\infty} T(\boldsymbol{x}) f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})\mathrm{d}\boldsymbol{x}.\end{aligned} \tag{24}$$

The gradient of the log-partition function with respect to the natural parameters yields the expected value of the sufficient statistics: $\nabla_{\boldsymbol{\theta}} A(\boldsymbol{\theta}) = \mathbb{E}_{p(x;\boldsymbol{\theta})}[T(x)]$ (Bishop, 2006). Then the mean of the sufficient statistic is followed as:

$$\int_{-\infty}^{\infty} T(\boldsymbol{x}) f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta})\mathrm{d}\boldsymbol{x} = \frac{\partial A^*(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \approx \frac{\partial A(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \qquad \because A^*(\boldsymbol{\theta}) = \boldsymbol{\theta}^{\intercal} A^*, \tag{25}$$

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^*. Plugging Eq. 25 into Eq. 23 then we have

$$\begin{aligned}\int_{-\infty}^{\infty} f_{\boldsymbol{x}}(\mathbf{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) \log f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) = -A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) + \boldsymbol{\theta}_{\hat{\boldsymbol{z}}}^{\intercal} \frac{\partial A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})}{\partial \boldsymbol{\theta}_{\hat{\boldsymbol{z}}}} \\ + \int_{-\infty}^{\infty} f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) B(\boldsymbol{x}) \mathrm{d}\boldsymbol{x},\end{aligned} \tag{26}$$

Applying same work to $\int_{-\infty}^{\infty} f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) \log f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}) \mathrm{d}\boldsymbol{x}$ and plug into Eq. 21 with Eq. 26, then the final Kullback-Leibler divergence of exponential family is followed as:

$$D_{\mathrm{KL}}(f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) \,\|\, f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})) = A(\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}) - A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}) + \boldsymbol{\theta}_{\hat{\boldsymbol{z}}}^{\intercal} \frac{\partial A(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}})}{\partial \boldsymbol{\theta}_{\hat{\boldsymbol{z}}}} - \boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}^{\intercal} \frac{\partial A(\boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}})}{\partial \boldsymbol{\theta}_{\hat{\boldsymbol{\epsilon}}}}. \tag{27}$$

C Objective Function

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

$$\begin{aligned}\mathcal{L}_{rec} &\coloneqq \frac{1}{k} \sum_{i=1}^{k} \left[\int q_i(\hat{\boldsymbol{z}}_i|\boldsymbol{x}) \log p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_i) \mathrm{d}\hat{\boldsymbol{z}}_i \prod_{j=1,\, j\neq i}^{k} \int q_j(\hat{\boldsymbol{z}}_j|\boldsymbol{x}) \mathrm{d}\hat{\boldsymbol{z}}_j \right] \\ &= \frac{1}{k} \sum_{i=1}^{k} E_{q_{\phi,\psi_i}(\boldsymbol{z}|\boldsymbol{x})} \log p_\theta(\boldsymbol{x}|\psi_i(\boldsymbol{z})),\end{aligned} \tag{28}$$

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

$$\log p_\theta(\boldsymbol{x}) = \int \prod_{i=1}^{k} q_i(\hat{\boldsymbol{z}}_i|\boldsymbol{x}) \log p_\theta(\boldsymbol{x}) \mathrm{d}\hat{\boldsymbol{z}}', \tag{29}$$

where $\mathrm{d}\hat{\boldsymbol{z}}' = \mathrm{d}\hat{\boldsymbol{z}}_1 \mathrm{d}\hat{\boldsymbol{z}}_2 \cdots \mathrm{d}\hat{\boldsymbol{z}}_k$. Expanding Eq. 29 as $\log p_\theta(\boldsymbol{x}) = \log \frac{p_\theta(\boldsymbol{x}, \hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \cdots, \hat{\boldsymbol{z}}_k)}{p_\theta(\hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \cdots, \hat{\boldsymbol{z}}_k|\boldsymbol{x})} = \log \frac{p_\theta(\boldsymbol{x}, \hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \cdots, \hat{\boldsymbol{z}}_k)}{q(\hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \cdots, \hat{\boldsymbol{z}}_k|\boldsymbol{x})} - \log \frac{p_\theta(\hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \cdots, \hat{\boldsymbol{z}}_k|\boldsymbol{x})}{q(\hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \cdots, \hat{\boldsymbol{z}}_k|\boldsymbol{x})}$. Then

$$\log p_\theta(\boldsymbol{x}) \geqq \int \prod_{i=1}^{k} q_1(\hat{\boldsymbol{z}}_i|\boldsymbol{x}) \log \frac{p_\theta(\boldsymbol{x}, \hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \cdots, \hat{\boldsymbol{z}}_k)}{q(\hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \cdots, \hat{\boldsymbol{z}}_k|\boldsymbol{x})} \mathrm{d}\hat{\boldsymbol{z}}'. \tag{30}$$

Expanding Eq. 30 then

$$\begin{aligned}\log p_\theta(\boldsymbol{x}) \geqq \int \prod_{i=1}^{k} q_i(\hat{\boldsymbol{z}}_i|\boldsymbol{x}) \Big[\log p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \cdots, \hat{\boldsymbol{z}}_k) \\ + \log \frac{p(\hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \cdots, \hat{\boldsymbol{z}}_k)}{q(\hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \cdots, \hat{\boldsymbol{z}}_k|\boldsymbol{x})} \Big] \mathrm{d}\hat{\boldsymbol{z}}'.\end{aligned} \tag{31}$$

The first term of Eq. 31 is expanded as:

$$p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \ldots, \hat{\boldsymbol{z}}_k) = \frac{p_\theta(\hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \ldots, \hat{\boldsymbol{z}}_k|\boldsymbol{x}) p_\theta(\boldsymbol{x})}{p_\theta(\hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \ldots, \hat{\boldsymbol{z}}_k)}. \tag{32}$$

Each IPE-transformation function operates independently, then all $\hat{\boldsymbol{z}}_i$ are independent and conditionally independent given $\boldsymbol{x}$. Then we have

$$p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \ldots, \hat{\boldsymbol{z}}_k) = \frac{p_\theta(\boldsymbol{x}) \prod_{i=1}^{k} p_\theta(\hat{\boldsymbol{z}}_i|\boldsymbol{x})}{\prod_{i=1}^{k} p_\theta(\hat{\boldsymbol{z}}_i)}, \tag{33}$$

where $\prod_{i=1}^{k} p_\theta(\hat{\boldsymbol{z}}_i|\boldsymbol{x}) = p_\theta(\hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \ldots, \hat{\boldsymbol{z}}_k|\boldsymbol{x})$ and $\prod_{i=1}^{k} p_\theta(\hat{\boldsymbol{z}}_i) = p_\theta(\hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \ldots, \hat{\boldsymbol{z}}_k)$. Let replace $p_\theta(\boldsymbol{x})$ as $p_\theta(\boldsymbol{x})^{-(k-1)} \prod_{i=1}^{k} p_\theta(\boldsymbol{x})$. Then we have

$$p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \ldots, \hat{\boldsymbol{z}}_k) = p_\theta(\boldsymbol{x})^{-(k-1)} \prod_{i=1}^{k} p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_i), \tag{34}$$

Plugging Eq. 34 into the first term of Eq. 31, then we have

$$\begin{aligned} &\int \prod_{i=1}^{k} q_i(\hat{\boldsymbol{z}}_i|\boldsymbol{x}) \log p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \cdots, \hat{\boldsymbol{z}}_k) \mathtt{d}\hat{\boldsymbol{z}}' \\ &= -(k-1) \log p_\theta(\boldsymbol{x}) + \int \prod_{i=1}^{k} q_i(\hat{\boldsymbol{z}}_i|\boldsymbol{x}) \Big[\sum_{j=1}^{k} \log p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_j) \Big] \mathtt{d}\hat{\boldsymbol{z}}'. \end{aligned} \tag{35}$$

Applying same process between the Eq. 32 and Eq. 33, and plugging Eq. 35 into Eq. 31, then we have

$$\begin{aligned} k \log p_\theta(\boldsymbol{x}) \geqq &\int \prod_{i=1}^{k} q_i(\hat{\boldsymbol{z}}_i|\boldsymbol{x}) \Big[\sum_{j=1}^{k} \log p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_j) \Big] \mathtt{d}\hat{\boldsymbol{z}}' \\ &+ \int \prod_{i=1}^{k} q_i(\hat{\boldsymbol{z}}_i|\boldsymbol{x}) \log \frac{\prod_{i=1}^{k} p(\hat{\boldsymbol{z}}_i)}{\prod_{i=1}^{k} q_i(\hat{\boldsymbol{z}}_i|\boldsymbol{x})} \mathtt{d}\hat{\boldsymbol{z}}'. \end{aligned} \tag{36}$$

Expanding the first term and inverse the second term of Eq. 36, then we have

$$\begin{aligned} \log p_\theta(\boldsymbol{x}) \geq &\frac{1}{k} \sum_{i=1}^{k} \left[\int q_i(\hat{\boldsymbol{z}}_i|\boldsymbol{x}) \log p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_i) \mathtt{d}\hat{\boldsymbol{z}}_i \prod_{j=1,\, j\neq i}^{k} \int q_j(\hat{\boldsymbol{z}}_j|\boldsymbol{x}) \mathtt{d}\hat{\boldsymbol{z}}_j \right] \\ &- \sum_{i=1}^{k} D_{\mathrm{KL}} \left(q_\phi(\hat{\boldsymbol{z}}_i|\boldsymbol{x}) \,\|\, p(\boldsymbol{z}_i) \right) \end{aligned} \tag{37}$$

Replacing $\int q_j(\hat{\boldsymbol{z}}_j|\boldsymbol{x}) \mathrm{d}\hat{\boldsymbol{z}}_j$ as 1, then we have:

$$\log p_\theta(\boldsymbol{x}) \geq \frac{1}{k} \left[\sum_{i=1}^{k} \mathbb{E}_{q_\phi(\hat{\boldsymbol{z}}_i|\boldsymbol{x})} \log p_\theta(\boldsymbol{x}|\hat{\boldsymbol{z}}_i) - k\, D_{\mathrm{KL}} \left(q_\phi(\hat{\boldsymbol{z}}_i|\boldsymbol{x}) \,\|\, p(\hat{\boldsymbol{z}}_i) \right) \right] \tag{38}$$

Therefore, we define ELBO as:

$$\begin{aligned} \mathcal{L}'(\phi, \theta, \psi_{i\in[1,k]}; \boldsymbol{x}) = &\frac{1}{k} \sum_{i=1}^{k} \mathbb{E}_{q_{\phi,\psi_i}(\boldsymbol{z}_i|\boldsymbol{x})} \log p_\theta(\boldsymbol{x}|\psi_i(\boldsymbol{z})) - \\ &\sum_{i=1}^{k} D_{\mathrm{KL}}(q_{\phi,\psi_i}(\boldsymbol{z}|\boldsymbol{x}) || p_{\psi_i}(\boldsymbol{z})). \end{aligned} \tag{39}$$

However, following Eq. 39, 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. 39 represents that increasing number of MIPE-transformation is equal to increasing β hyper-parameter in β-VAE (Higgins et al., 2017).

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

$$\mathcal{L}(\phi, \theta, \psi_{i\in[1,k]}; \boldsymbol{x}) = \mathcal{L}_{rec} - \mathcal{L}_{kl} - \mathcal{L}_{el} - \mathcal{L}_{cali}. \tag{40}$$

D Details of Experimental Environment

D.1 Model Architecture and Hyper-Parameter Tuning

The details of model architecture is demonstrated in Table 6and 7. The details of hyper-parameter settings are in Table 8.

Table 6: VAE architecture for dSprites dataset.

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. $4 \times 4 \times 64$ 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 7: VAE architecture for 3D Shapes, and 3D Cars datasets. For exceptional case, CLG-VAE, we use ten dimension size on 3D Shapes dataset (Zhu et al., 2021).

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

Table 8: Hyper-parameters for dSprites, 3D Cars, and 3D Shapes. The epochs for dSprites and 3D cars are 30 and 200, respectively. lr is learning rate, latent dim is latent vector size, group reconst is group reconstruction, and forward group is forward group pass.

(a) Hyper-parameters for dSprites and 3D Cars

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

(b) Hyper-parameters for 3D Shapes

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

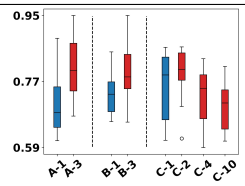
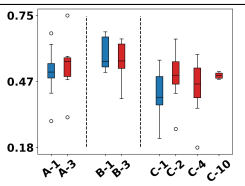
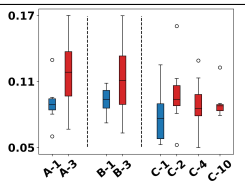
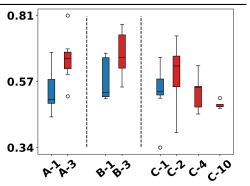
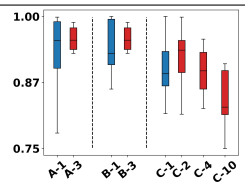
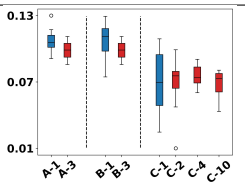
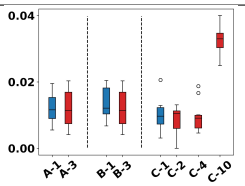
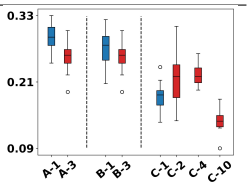
D.2 Details of Datasets

We benchmark MIPET-VAEs against several standard VAEs on the following datasets: 1) dSprites (Matthey et al., 2017) which consists of 737,280 binary 64×64 images of dSprites with five independent ground truth factors(number of values), *i.e.* shape(3), orientation(40), scale(6), x-position(32), and y-position(32). 2) 3D Shapes (Burgess and 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).

D.3 Details of Training and Evaluation

Training We set 256 mini-batch size in the datasets (dSprites, 3D Shapes, and 3D Cars), Adam optimizer with learning rate 4×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and epochs from $\{30, 67, 200\}$ as a common setting for all the comparative methods. For the comparison, we follow training and inference on the whole dataset. We train each model for 30, 67, and 200 epochs on the dSprites, 3D Shapes, and 3D Cars, respectively, as introduced in Kim and Mnih (2018); Ren et al. (2022). We tune β from $\{1, 2, 4, 10\}$ and $\{4, 6\}$ for β-VAE and β-TCVAE, respectively.

Table 9: Impact of the number of MIPE-transformation function on the β-TCVAE and β-VAE with dSprites, 3D Shapes, and 3D Cars datasets in terms of the four metrics. The blue and red box plots represent each model's single and multiple IPE-transformation cases, respectively. (A-n: MIPET-β-TCVAE (4), B-n: MIPET-β-TCVAE (6), C-n: MIPET-β-VAE, n: the number of MIPE-transformation).

dataset	Metrics			
	FVM	MIG	SAP	DCI
3D Shapes				
3D Cars				

We apply the proposed method to β-TCVAE only with the EF similarity loss term because β-TCVAE penalizes the divided KL divergence terms. We set the same encoder and decoder architecture in each model to exclude the overlapped effects. The $\alpha = 1$ and $\gamma = 1$ of β-TCVAE as Chen et al. (2018). 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 λ_{decomp}, λ_{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 $\{0.2, 0.5, 0.7\}$. For Control-VAE, we set the maximum KL divergence value from $\{10, 12, \ldots, 20\}$. In addition, we set masking ratio λ from $\{0.0, 0.5, \ldots 2.0, \infty\}$. To check the impact of MIPE-transformation, we do not consider the Groupified VAE because the latter is implemented with an extended decoder (different capacity).

Evaluation We conduct experiments on NVIDIA A100, RTX 2080 Ti, and RTX 3090. We set 100 samples to evaluate global empirical variance in each dimension and run it a total of 800 times to estimate the FVM score introduced in Kim and Mnih (2018). For the MIG (Chen et al., 2018), SAP (Kumar et al., 2018), and DCI (Eastwood and Williams, 2018), we follow default values introduced in Michlo (2021), training and evaluation 100 and 50 times with 100 mini-batches, respectively. We evaluate four disentanglement metrics for a less biased understanding of the actual states of disentanglement.

D.4 Additional Result of Relation Between Factors and Latent Vector Dimensions

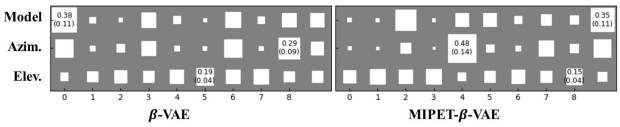


Figure 6: 3D Car Dataset: Azim. refers to the azimuth factor, and Elev. refers to the elevation factor.

In the 3D Car dataset, a trade-off was observed. While the maximum value across models and evaluation factors showed a slight decrease, the azimuth factor exhibited a notable improvement compared to the declines seen in other factors.

D.5 Impact of Symmetric Matrix Exponential

We empirically show the benefit of using a symmetric matrix for ψ. Table 10 shows the ratio of runs with a symmetric matrix, which shows better performance than unrestricted matrices, to the total 240 (60 models $\times$ 4

Table 10: The ratio of seeds to show better performance with symmetric matrix

dSprites	3D Shapes	3D Cars
0.58	0.56	0.67

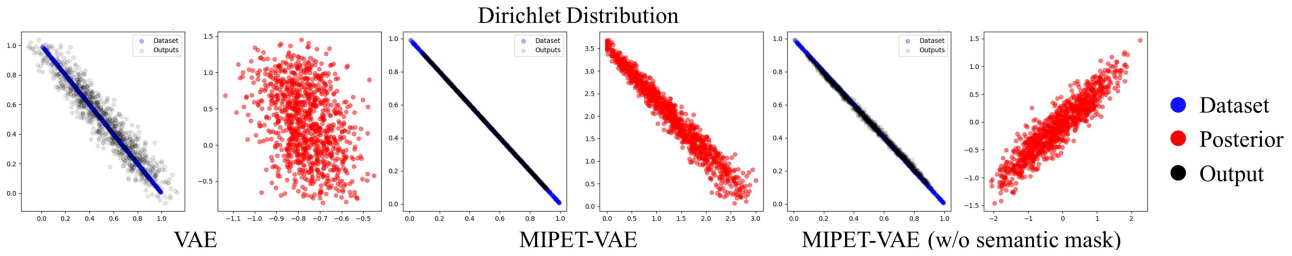


Figure 7: 2D Exponential Family Distribution Datasets

metrics) runs for each dataset. All results are higher than 0.5, which implies that the constraint enhances I2L equivariance even with uncertain factors.

D.6 Additional Experiment of Computing Complexity

Table 11: Training complexity.

# of IE	Complexity
0	$\times$ 1.00
1	$\times$ 0.75
3	$\times$ 0.50
4	$\times$ 0.33

We additionally estimate the computing complexity depending on the number of IPE-transformation. The results are in Table 11 and represent the training time complexity compare to baselines (when the number of IE is equal to 0).

D.7 Semantic Mask: Mapping Gaussian Distribution to Diverse Exponential Family

As shown in Fig. 7, MIPET-VAE with semantic mask preserves the dataset structure compared to VAE. Also, EF conversion with semantic mask maps the Gaussian distribution to the exponential family compared to MIPET without the semantic mask and VAE cases.