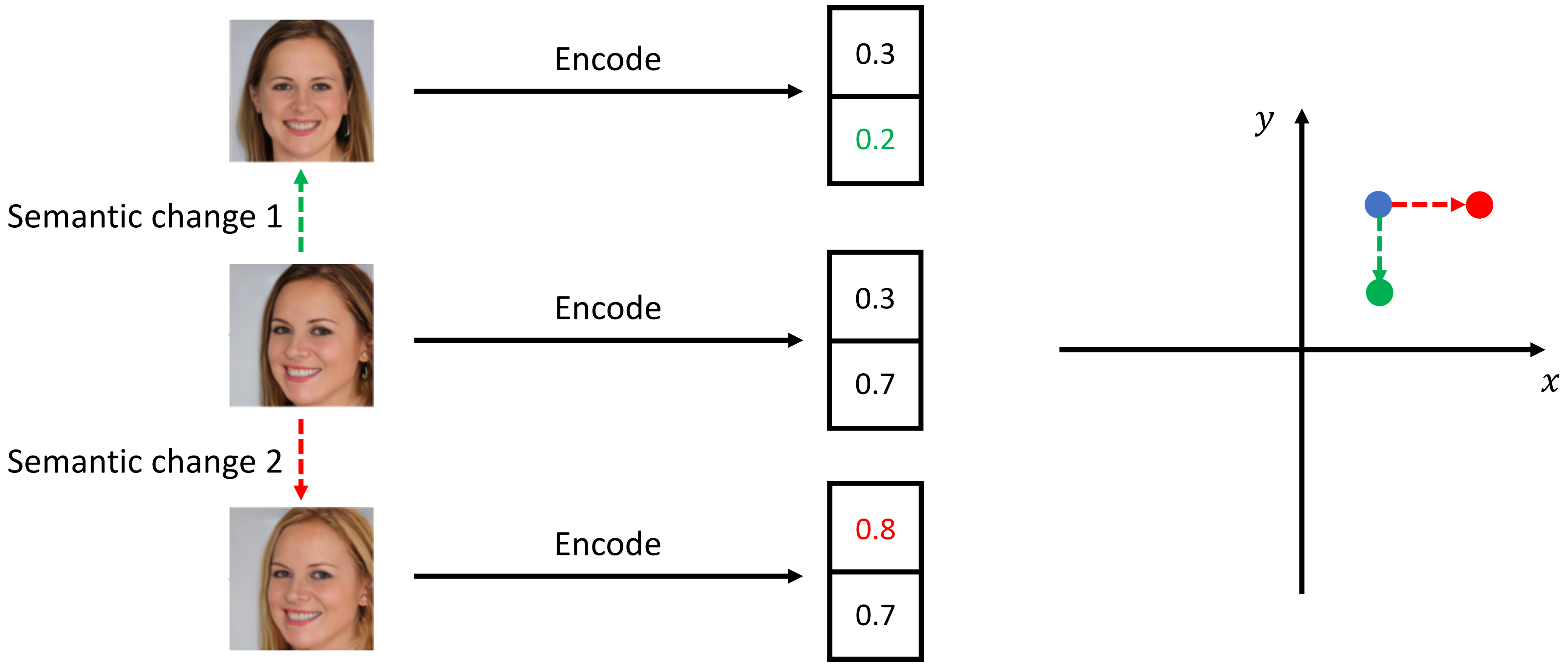
Commutative Lie Group VAE for Disentanglement Learning (Long Talk)

Xinqi Zhu, Chang Xu, Dacheng Tao

The University of Sydney

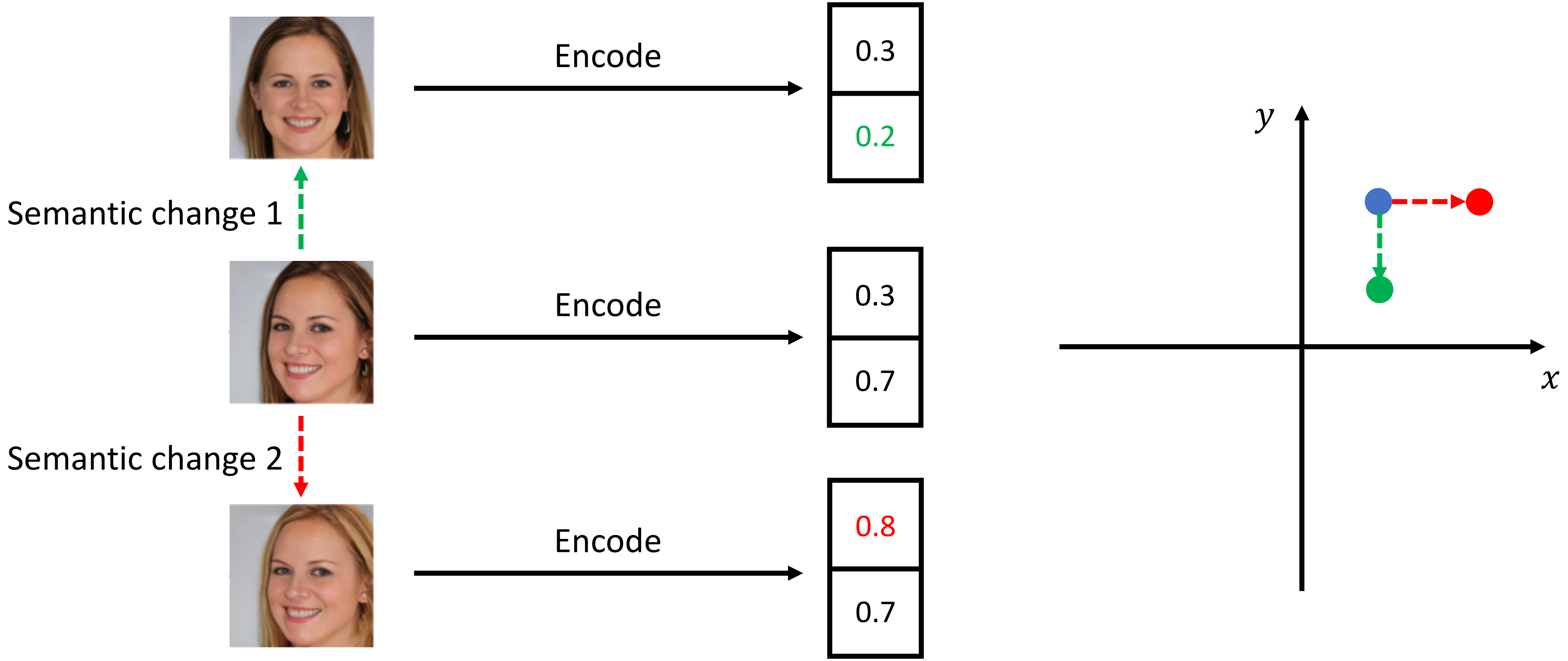
Rethinking Disentanglement Learning



Traditional disentangled representation.

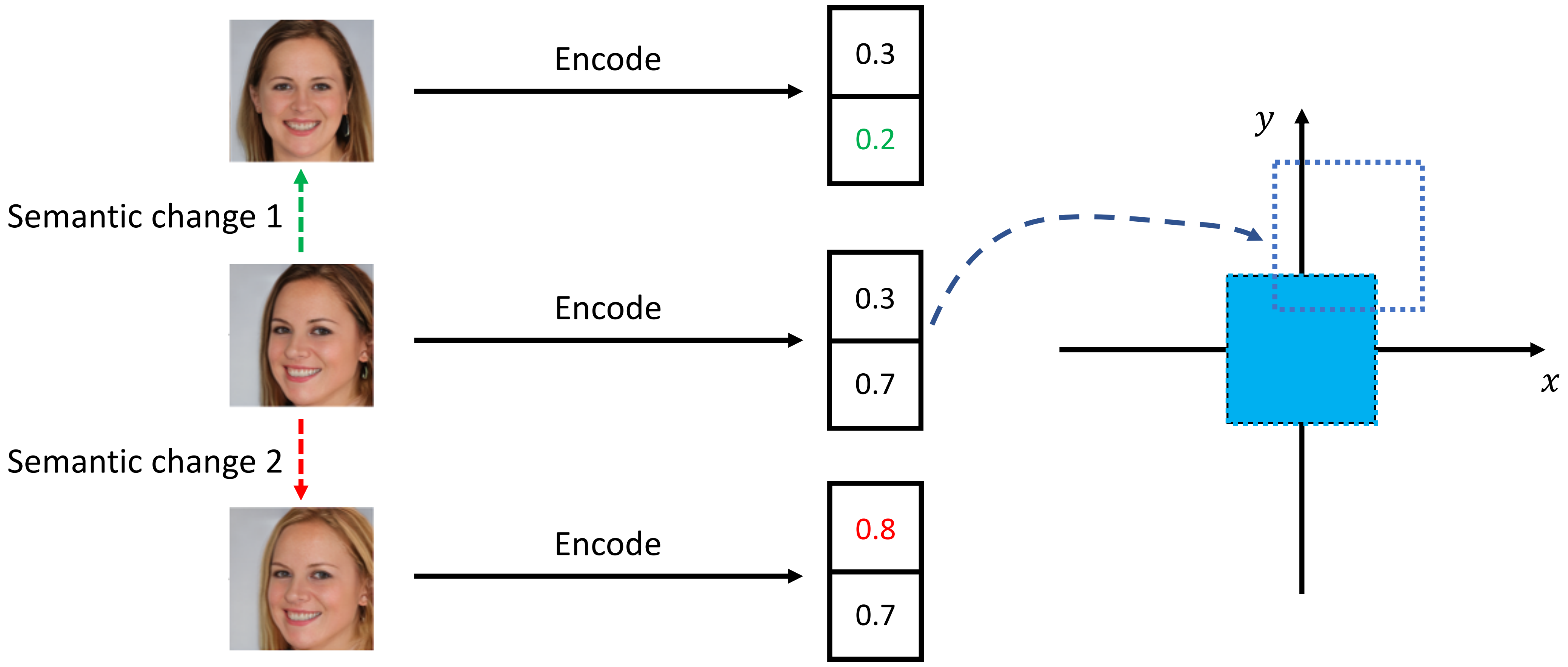
We view each embedded sample as a point.

Rethinking Disentanglement Learning



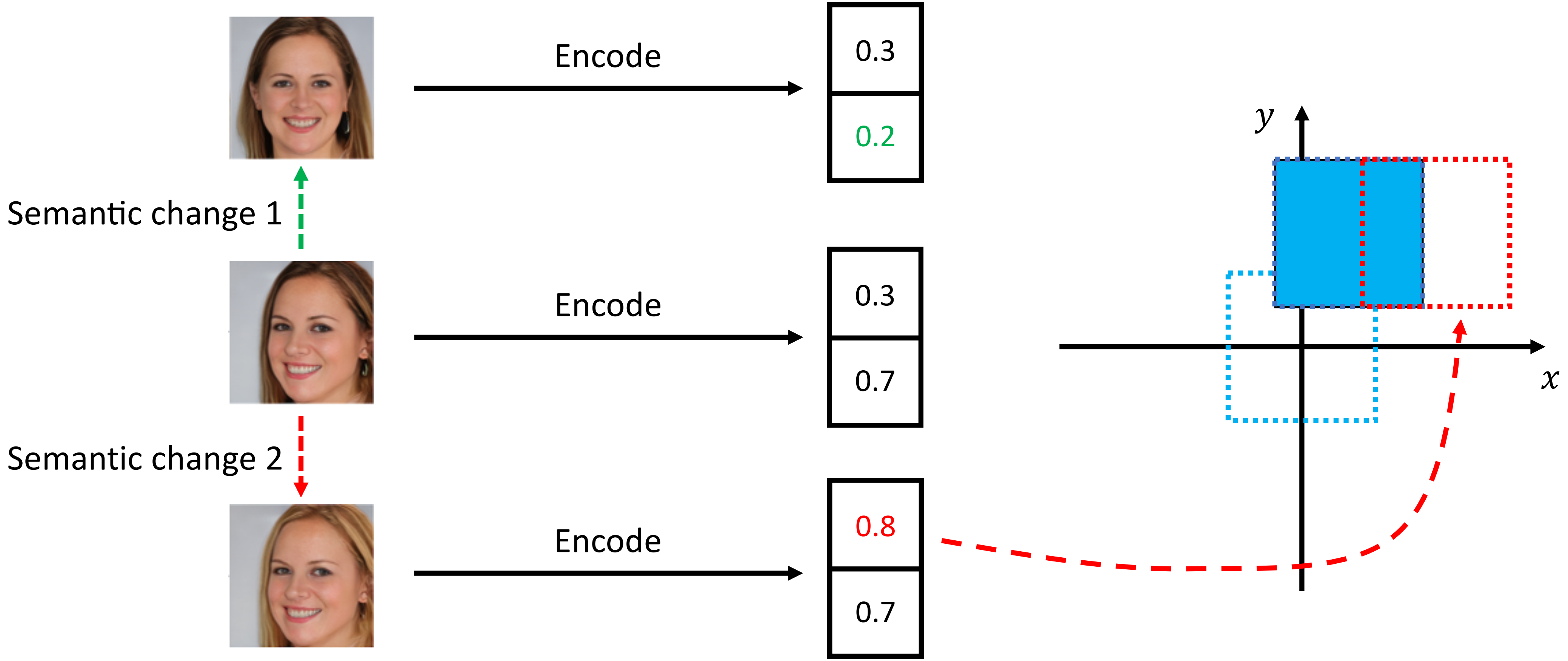
However, we can also view it as space translation.

Rethinking Disentanglement Learning



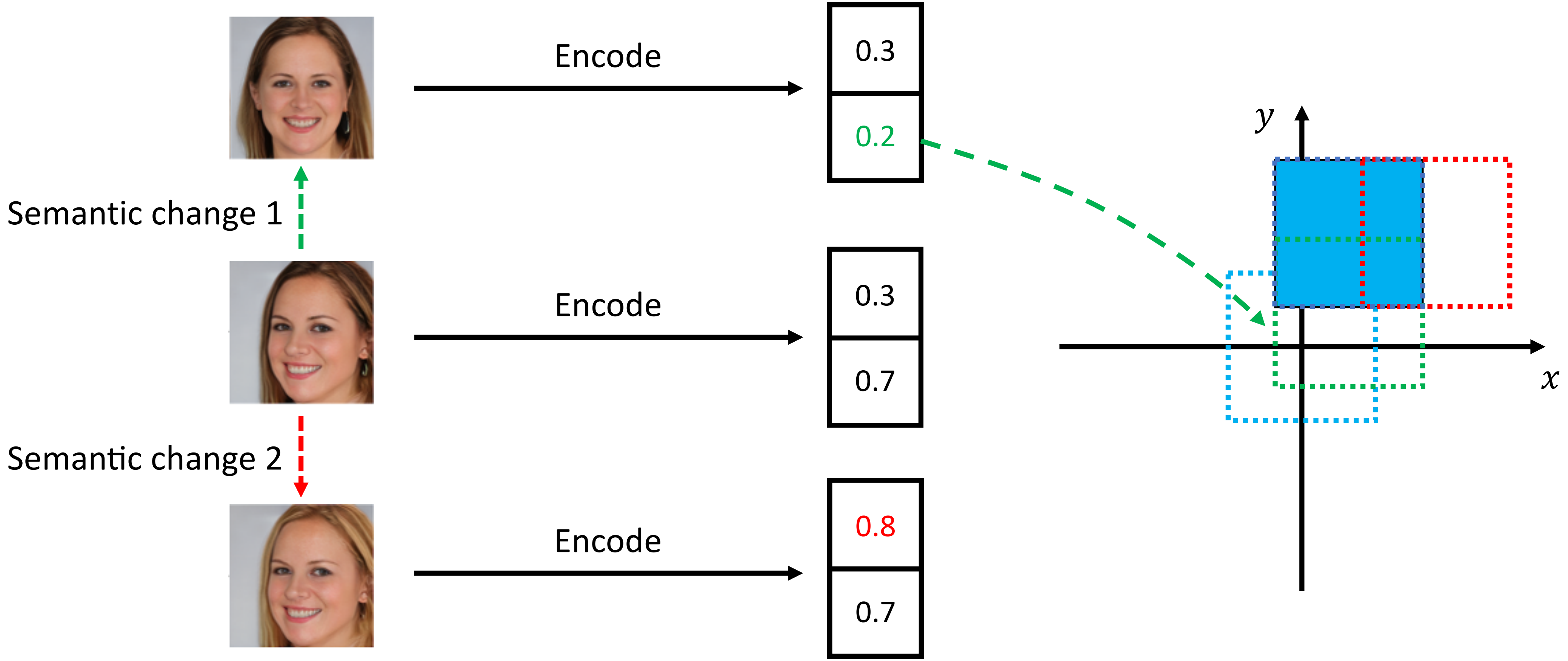
However, we can also view it as space translation.

Rethinking Disentanglement Learning



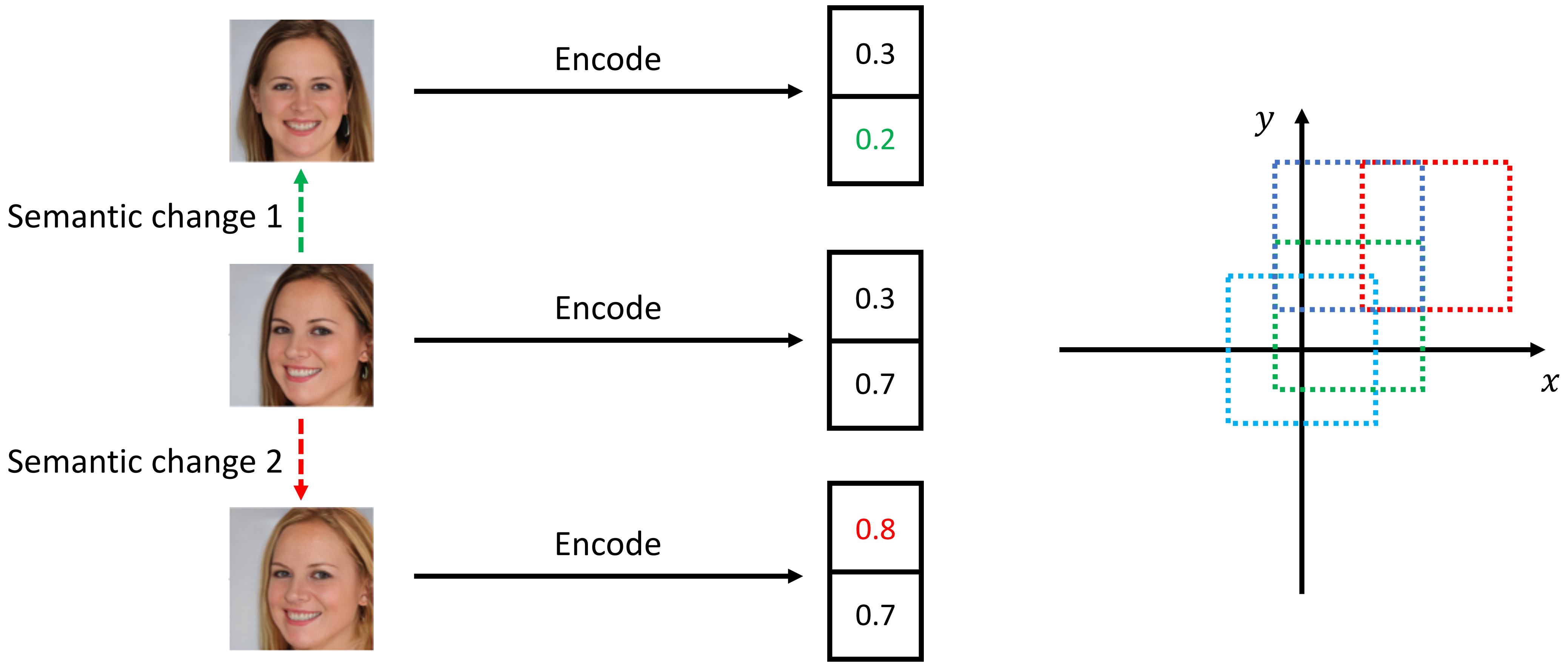
However, we can also view it as space translation.

Rethinking Disentanglement Learning



However, we can also view it as space translation.

Rethinking Disentanglement Learning



However, we can also view it as space translation.

Image variation $\xleftrightarrow{\text{Equivariance}}$ Latent space translation

Rethinking Disentanglement Learning

Image variation $\xleftrightarrow{\text{Equivariance}}$ Latent space $\boxed{\text{translation}}$

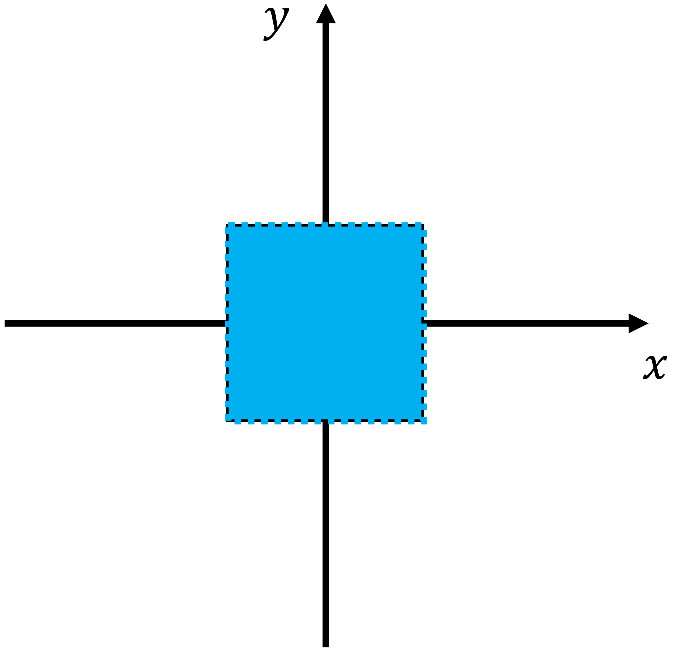
Why translation?

Other transformations are possible.

Rethinking Disentanglement Learning

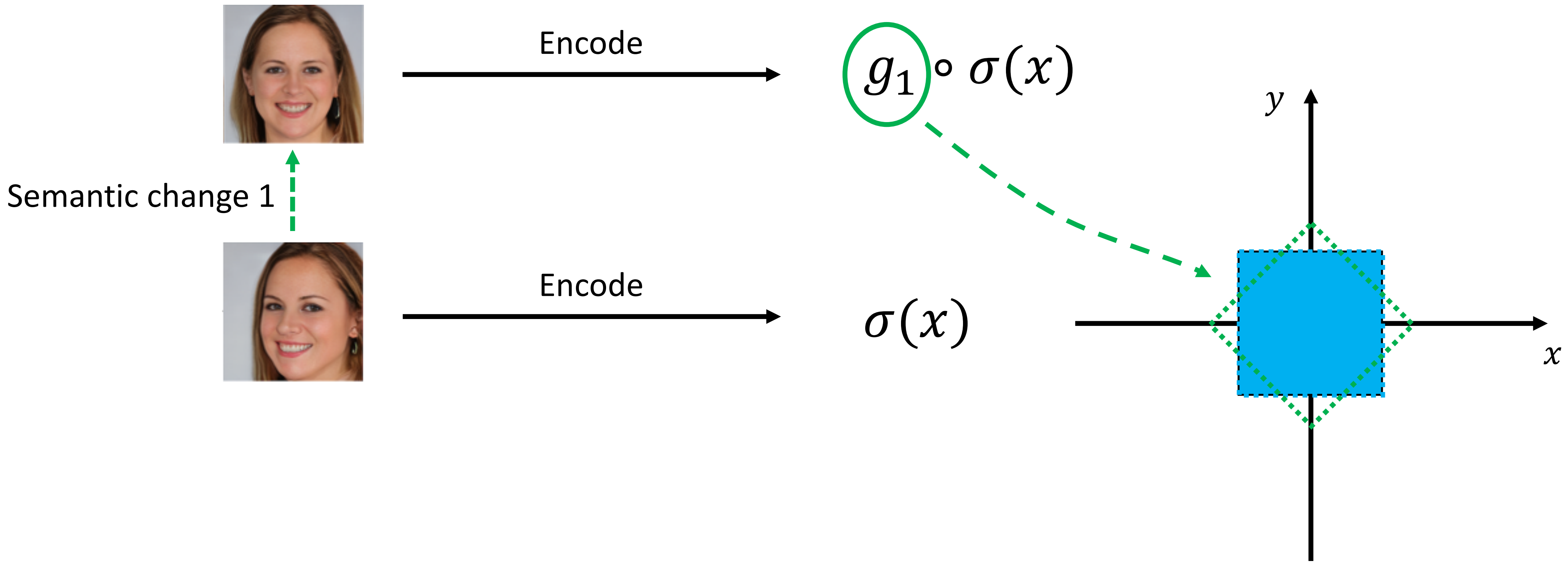
Encode

$\sigma(x)$



Suppose we encode an image as a state of the latent space.

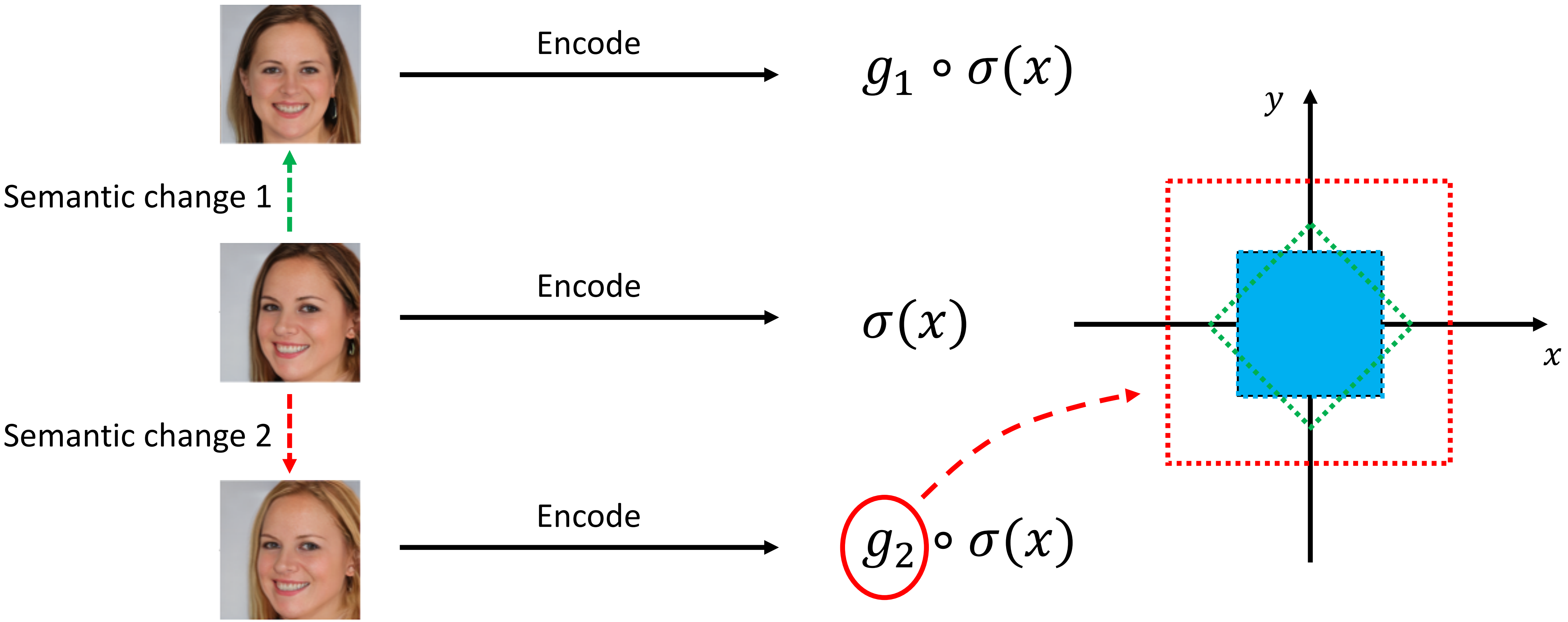
Rethinking Disentanglement Learning



Suppose we encode an image as a state of the latent space.

The different image changes now correspond to different latent transformations.

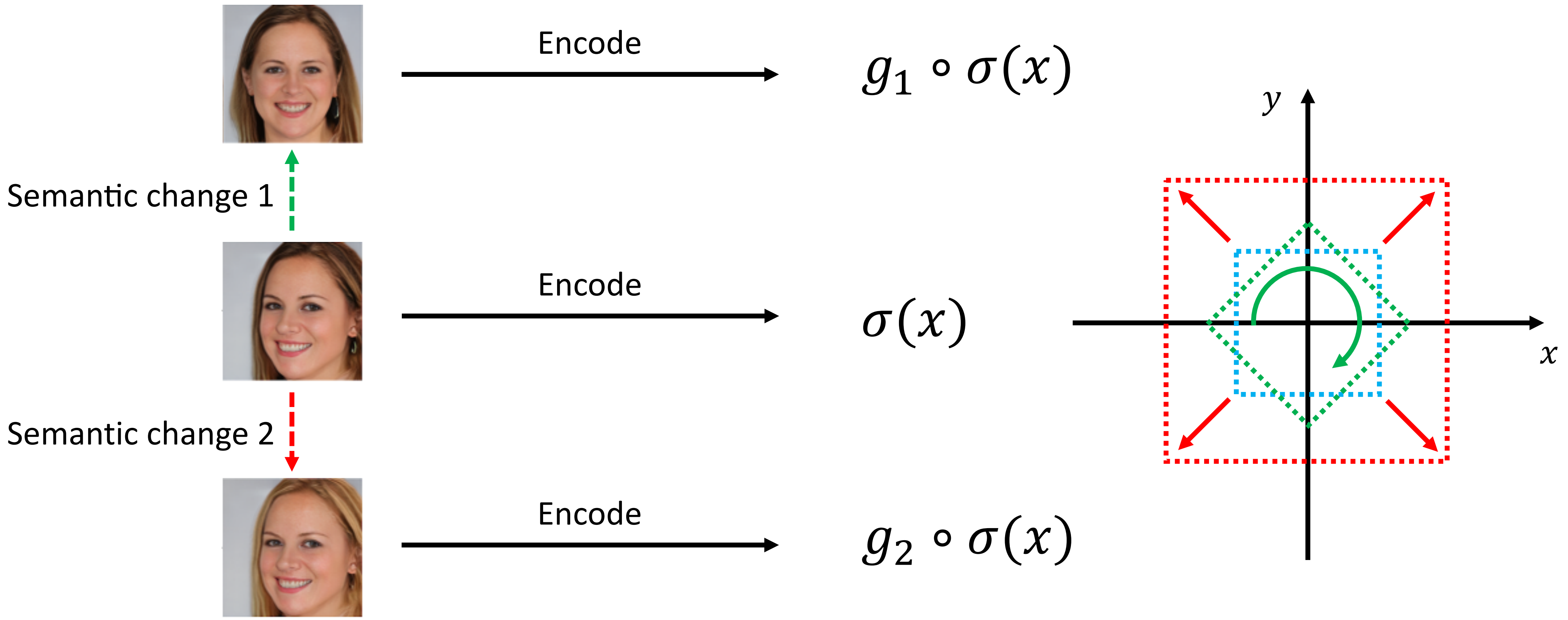
Rethinking Disentanglement Learning



Suppose we encode an image as a state of the latent space.

The different image changes now correspond to different latent transformations.

Rethinking Disentanglement Learning



Semantic 1 $= g_1$: rotating transformation.

Semantic 2 $= g_2$: scaling transformation.

Rethinking Disentanglement Learning

A brief comparison:

Old fashion:

Semantic 1 $= g_1$: dim-1 translation.

Semantic 2 $= g_2$: dim-2 translation.

New fashion:

Semantic 1 $= g_1$: rotating transformation.

Semantic 2 $= g_2$: scaling transformation.

OK, but why do we need the new fashion?

Rethinking Disentanglement Learning

A brief comparison:

These translations are fixed and predefined.

Old fashion:

Semantic 1 $= g_1$: dim-1 translation.

Semantic 2 $= g_2$: dim-2 translation.

New fashion:

Semantic 1 $= g_1$: rotating transformation.

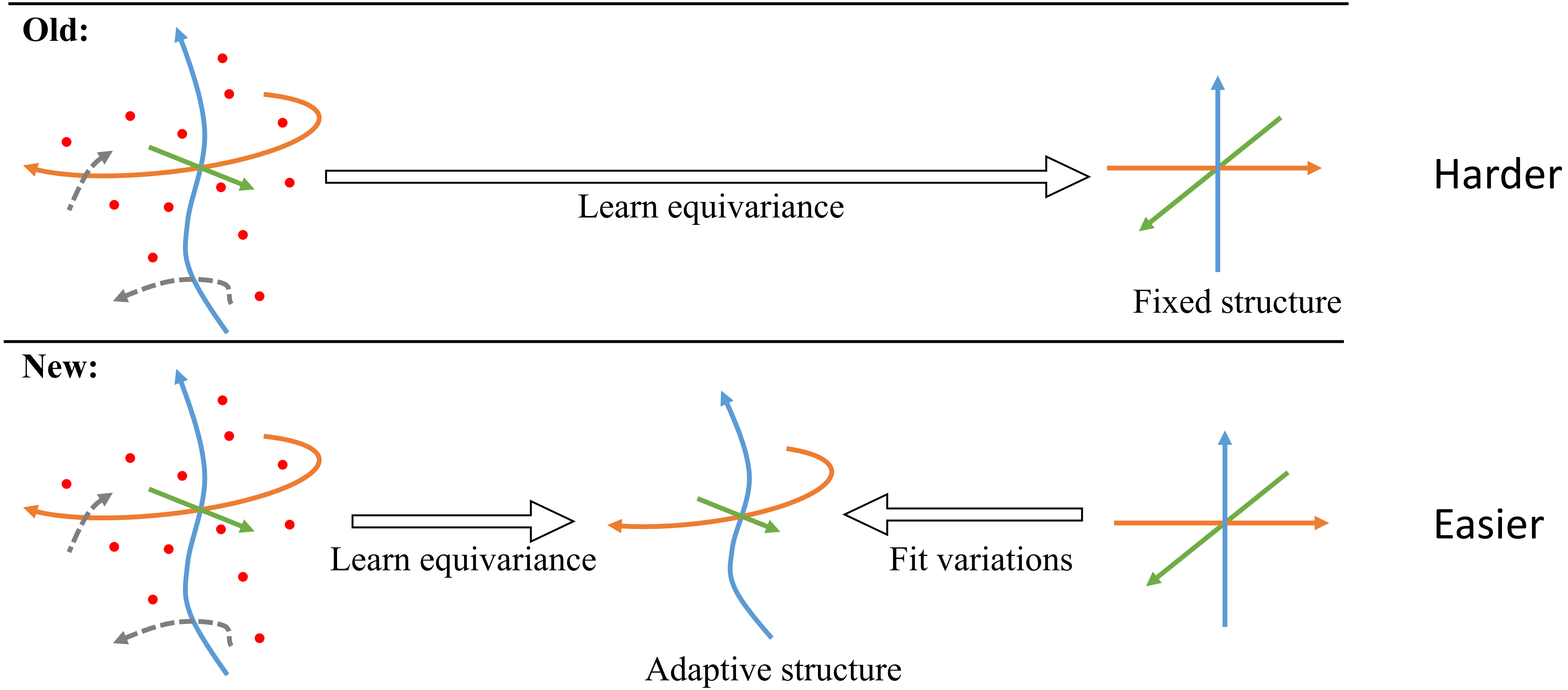
Semantic 2 $= g_2$: scaling transformation.

These transformations can be learned adaptively!

Rethinking Disentanglement Learning

Hypothesis:

The target equivariance can be more easily learned if an adaptive transformation structure is used to capture the semantic variations than a fixed structure.



In this work, we use a learnable group structure to achieve this goal.

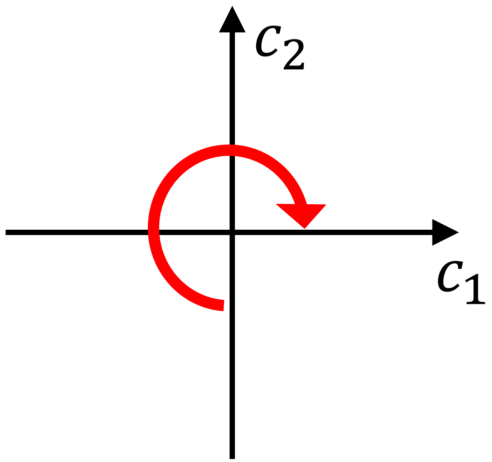
And in this work, we only consider continuous matrix groups.

An Impression on Group

A group element

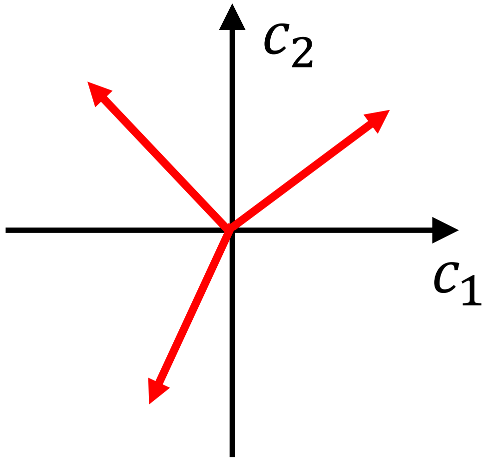
$$\sigma'(x) = g \circ \sigma(x)$$

$$\begin{bmatrix} c_1' \\ c_2' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$



Or

$$\begin{bmatrix} c_1' \\ c_2' \end{bmatrix} = \begin{bmatrix} S_\theta & 0 \\ 0 & S_\theta \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$



The group element $g \in G$ maps a vector space to itself: $\mathbb{R}^2 \to \mathbb{R}^2$.

An Impression on Group

A group element

$$\sigma'(x) = g \circ \sigma(x)$$

$$\begin{bmatrix} c_1' \\ c_2' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

$$\begin{bmatrix} \cos\phi' & \sin\phi' \\ -\sin\phi' & \cos\phi' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix}$$

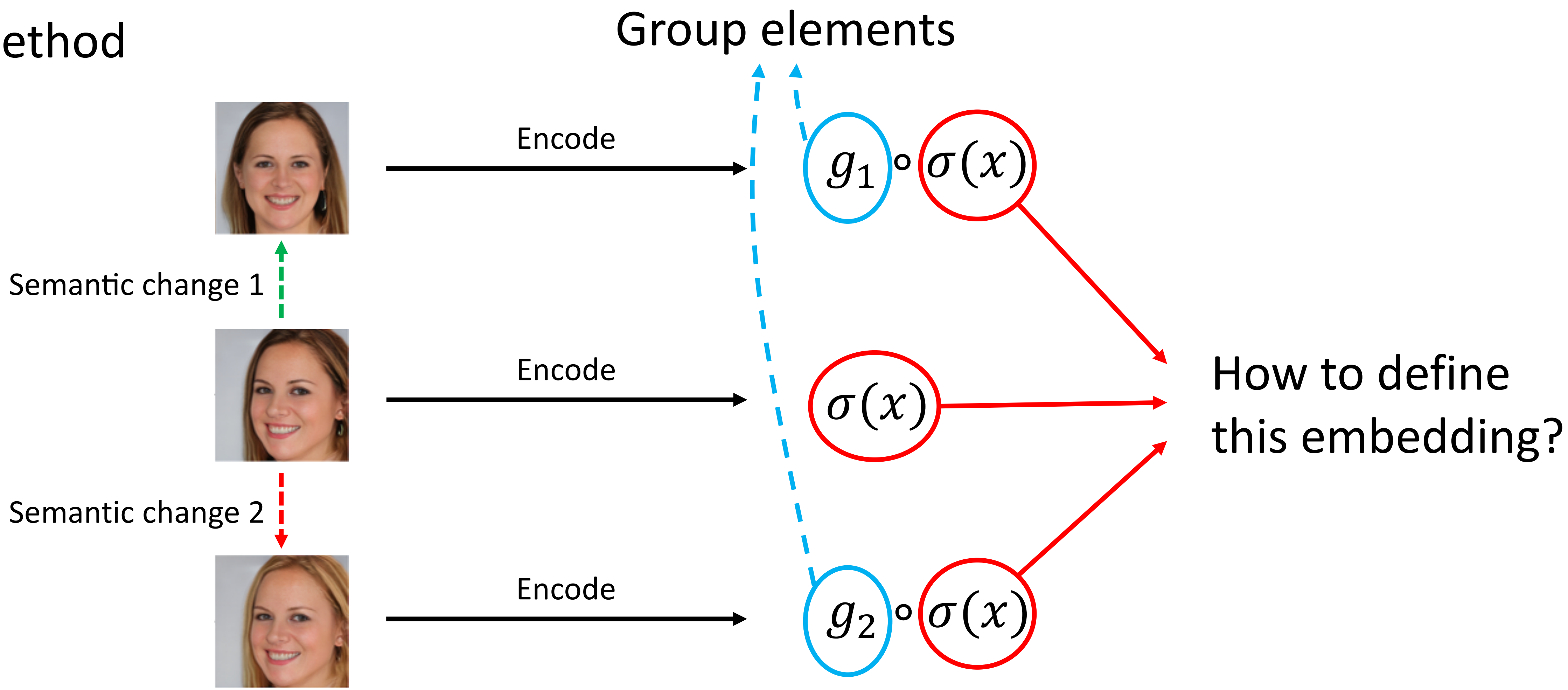
Or

$$\begin{bmatrix} c_1' \\ c_2' \end{bmatrix} = \begin{bmatrix} S_\theta & 0 \\ 0 & S_\theta \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

$$\begin{bmatrix} S_{\phi'} & 0 \\ 0 & S_{\phi'} \end{bmatrix} = \begin{bmatrix} S_\theta & 0 \\ 0 & S_\theta \end{bmatrix} \begin{bmatrix} S_\phi & 0 \\ 0 & S_\phi \end{bmatrix}$$

Now the group element $g \in G$ maps a group structure to itself: $G \to G$.

Method



We assume there is a canonical sample x_0, and every other sample is transformed from the canonical one:

$$\sigma(x) = g_{0\to x} \circ \sigma(x_0),$$

and:

$$g_1 \circ \sigma(x) = (g_1 \circ g_{0\to x}) \circ \sigma(x_0);$$

$$g_2 \circ \sigma(x) = (g_2 \circ g_{0\to x}) \circ \sigma(x_0).$$

Method: (1) Group representation

We assume there is a canonical sample x_0, and every other sample is transformed from the canonical one:

$$\sigma(x) = g_{0\to x} \circ \boxed{\sigma(x_0)}, \quad \text{and:} \quad \begin{aligned} g_1 \circ \sigma(x) &= (g_1 \circ g_{0\to x}) \circ \boxed{\sigma(x_0)}; \\ g_2 \circ \sigma(x) &= (g_2 \circ g_{0\to x}) \circ \boxed{\sigma(x_0)}. \end{aligned}$$

It's the group $g \in G$ that defines the representation structure (relation between samples).

The canonical embedding $\sigma(x_0)$ can be seen as a constant.

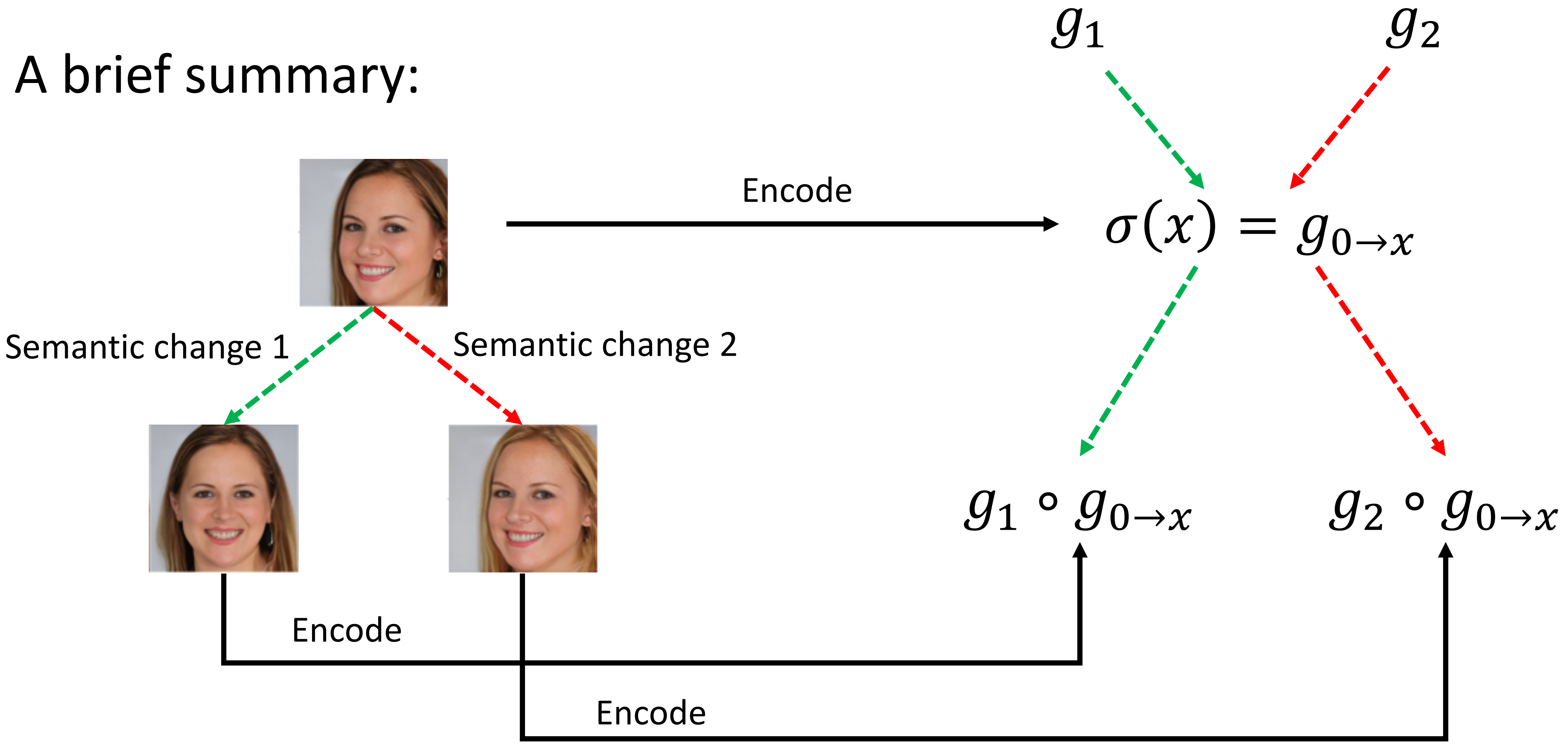
We propose to set $\sigma(x_0)$ to be a fixed value: the group identity element (e).

$$\sigma(x) = g_{0\to x} \circ \sigma(x_0) = g_{0\to x} \circ e = g_{0\to x},$$

Now the samples are embedded on a group structure.
We name this embedding as the 'group representation'.

Method: (1) Group representation

A brief summary:



…looks cool, but how to disentangle $\sigma(x) = g_{0\to x}$?

Method: (1) Group representation

Expectation:

$$\sigma(x) = g_{0\to x} = \underbrace{g_1(x)}_{\text{Semantic 1}} \circ \underbrace{g_2(x)}_{\text{Semantic 2}} \ldots \circ \underbrace{g_m(x)}_{\text{Semantic m}} \circ \underbrace{e}_{\text{Canonical}}$$

We expect every embedding to be decomposed into subgroup actions.

When a new semantic change comes:

$$g_2 \circ \sigma(x) = g_2 \circ g_{0\to x} = g_1(x) \circ (g_2 \circ g_2(x)) \ldots \circ g_m(x) \circ e$$

Method: (1) Group representation

Expectation:

$$\sigma(x) = g_{0\to x} = \underbrace{g_1(x)}_{\text{Semantic 1}} \circ \underbrace{g_2(x)}_{\text{Semantic 2}} \ldots \circ \underbrace{g_m(x)}_{\text{Semantic m}} \circ \underbrace{e}_{\text{Canonical}}$$

In this case, disentanglement on the group representation is achieved via subgroup decomposition.

…looks cool, but how to learn this decomposable group representation?

Method: (2) Lie Algebra Parameterization

$$\sigma(x) = g_{0\rightarrow x}$$

This representation is on a group structure.

At lease we need a way to obtain elements on a group!

In this work, we focus on Lie group and adopt Lie algebra parameterization:

$$g(t) = \exp\big(A(t)\big), g \in G, A \in \mathfrak{g},$$

$$A(t) = t_1 A_1 + t_2 A_2 + \cdots + t_m A_m, \forall t_i \in \mathbb{R}, A_i \in \mathbb{R}^{d\times d}.$$

Basis

Coordinates

Method: (2) Lie Algebra Parameterization

$$\sigma(x) = g_{0\to x}$$

This representation is on a group structure.

At lease we need a way to obtain elements on a group!

In this work, we focus on Lie group and adopt Lie algebra parameterization:

$$g(t) = \exp\big(A(t)\big), g \in G, A \in \mathfrak{g},$$
$$A(t) = t_1 A_1 + t_2 A_2 + \cdots + t_m A_m, \forall t_i \in \mathbb{R}, A_i \in \mathbb{R}^{d\times d}.$$

Now, a group G is defined through a vector space A.

Method: (2) Lie Algebra Parameterization

Since the Lie algebra is a vector space, we can now use general optimization methods (e.g. SGD, Adam) to learn a group structure!

We view the basis $\{A_i\}_{i=1}^{m}$ as learnable weights in a deep model as it determines the structure of the group.

Method: (2) Lie Algebra Parameterization

In this work, we enforce a straightforward group decomposition, namely 'one-parameter subgroup decomposition':

$$
\begin{aligned}
& g(t) \\
& = \exp(t_1 A_1 + t_2 A_2 + \cdots + t_m A_m) \\
& = \exp(t_1 A_1) \exp(t_2 A_2) \ldots \exp(t_m A_m)
\end{aligned}
$$

$g_1(t_1)$	$g_2(t_2)$	$g_m(t_m)$
Semantic 1	Semantic 2	Semantic m

Unfortunately, this decomposition doesn't hold in general.

Method: (3) Disentangle via Decomposition

A proposition is proposed to enforce this decomposition:

Proposition 1. *If* $A_i A_j = A_j A_i, \forall i, j,$ *then*

$$\begin{aligned}
&\exp(t_1 A_1 + t_2 A_2 + \ldots t_m A_m) \\
&= \exp(t_1 A_1)\exp(t_2 A_2)\ldots\exp(t_m A_m) \\
&= \prod_{\mathrm{perm}(i)} \exp(t_i A_i).
\end{aligned}$$

Proof. See Appendix 1.

We can see this group decomposition is commutative, and it is where the 'commutative' in the title comes from.

Method: (3) Disentangle via Decomposition

Furthermore, we also consider another disentanglement constraint called Hessian Penalty (Peebles et al., 2020) on the group structure:

Proposition 2. *If* $A_i A_j = 0, \forall i \neq j$, *then*

$$H_{ij} = \frac{\partial^2 g(t)}{\partial t_i \partial t_j} = 0,$$

where g is the map defined in Eq. 4.

Proof. See Appendix 2.

This is a stronger constraint than Prop. 1 since $A_i A_j = A_j A_i$ is implied by $A_i A_j = 0$.

Method: (4) Constructing a VAE Model

Before imposing the proposed disentanglement constraints, we first introduce a VAE variant called bottleneck-VAE:

Proposition 3. *Suppose two latent variables z and t are used to model the log-likelihood of data x, then we have:*

$$\begin{aligned} \log p(x) &\geq \mathcal{L}_{bottleneck}(x, z, t) \\ &= \mathbb{E}_{q(z|x)}\mathbb{E}_{q(t|x,z)}\log p(x, z|t) \\ &\quad - \mathbb{E}_{q(z|x)}KL(q(t|x,z)||p(t)) - \mathbb{E}_{q(z|x)}\log q(z|x) \quad (9) \\ &= \mathbb{E}_{q(z|x)q(t|z)}\log p(x|z)p(z|t) \\ &\quad - \mathbb{E}_{q(z|x)}KL(q(t|z)||p(t)) - \mathbb{E}_{q(z|x)}\log q(z|x), \quad (10) \end{aligned}$$

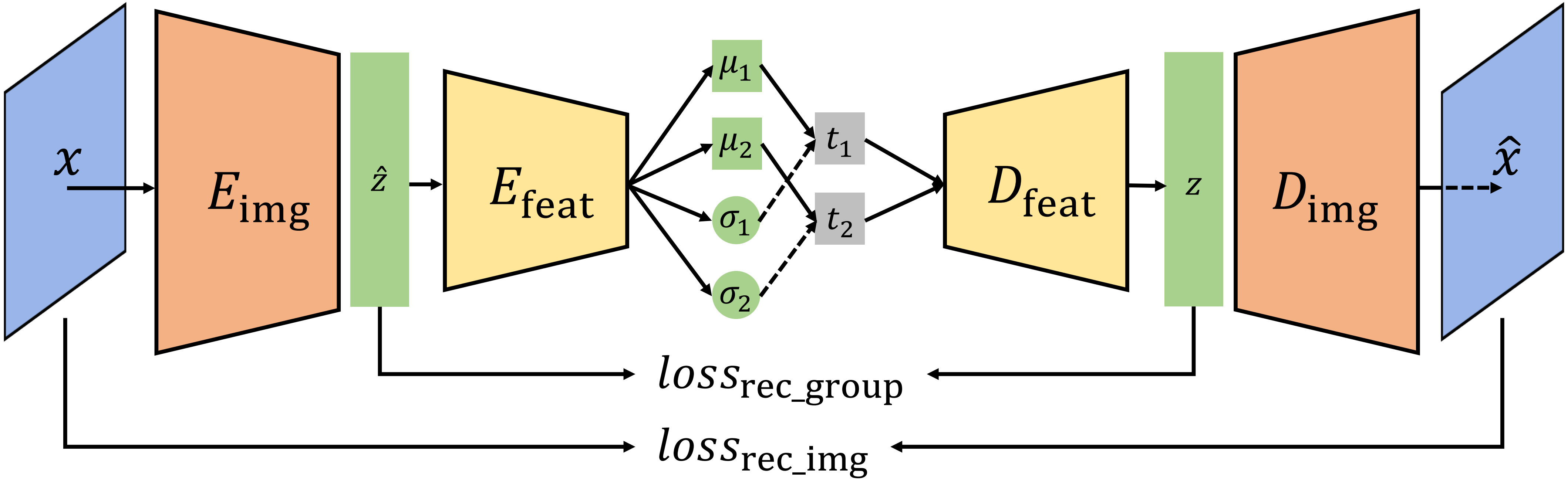
where Eq. 10 holds because we assume Markov property: $q(t|z) = q(t|x, z)$, $p(x|z, t) = p(x|z)$.

Proof. See Appendix 4. □

Method: (4) Constructing a VAE Model

Prop. 3 defines a VAE variant which shares a layer of feature in the encoder and the decoder:

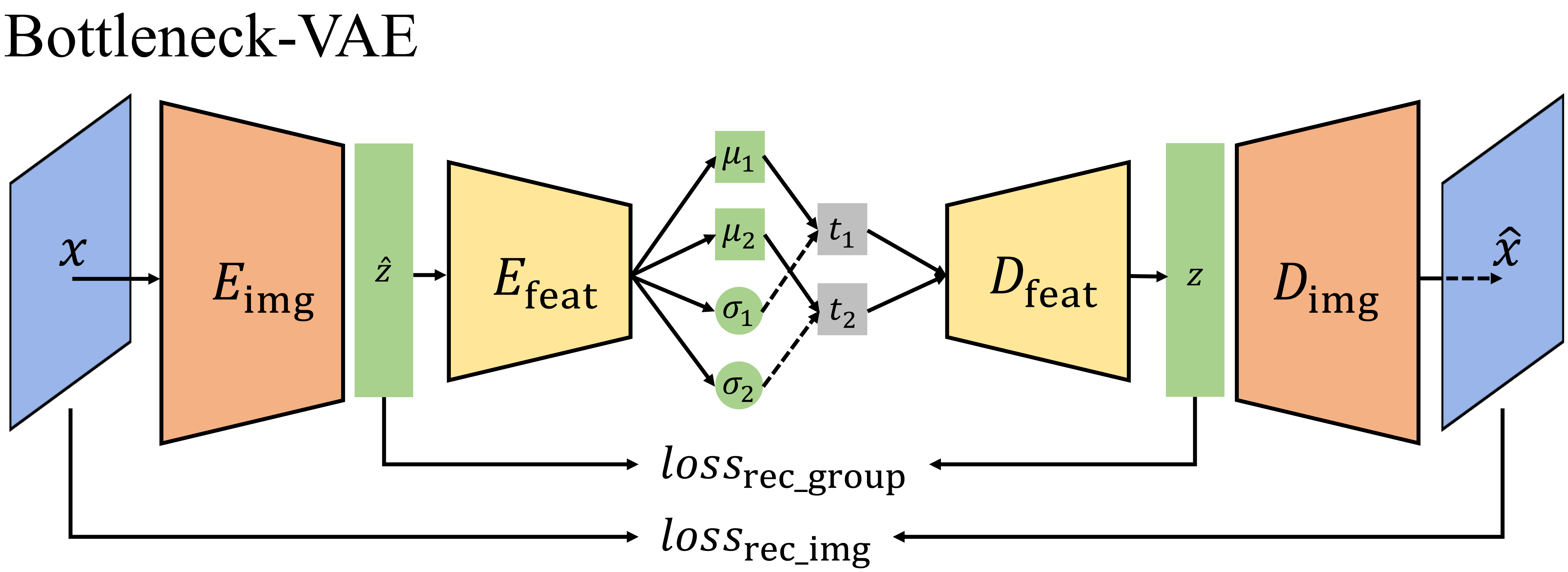
Bottleneck-VAE



In addition to a standard VAE, this model enforce z and $\hat{z}$ to be equal.

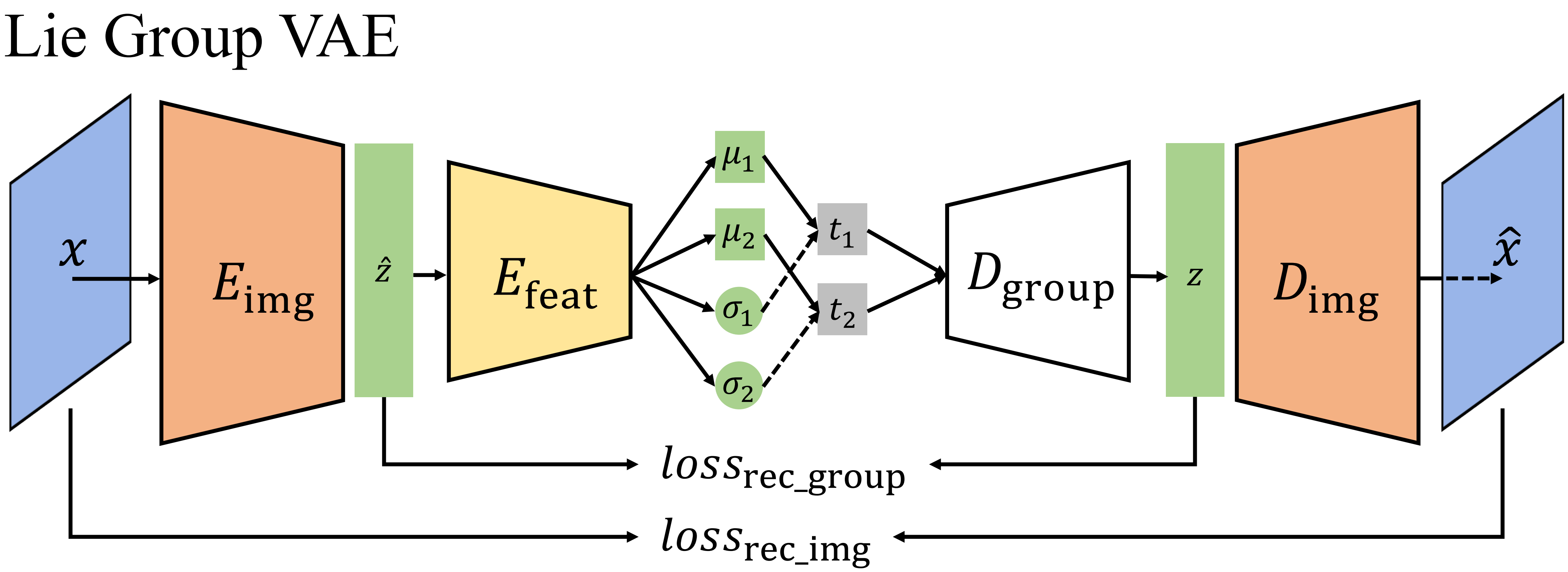
Method: (4) Constructing a VAE Model

Our proposed Lie Group VAE is a slight variant of bottleneck-VAE, which has a special group feature decoder:



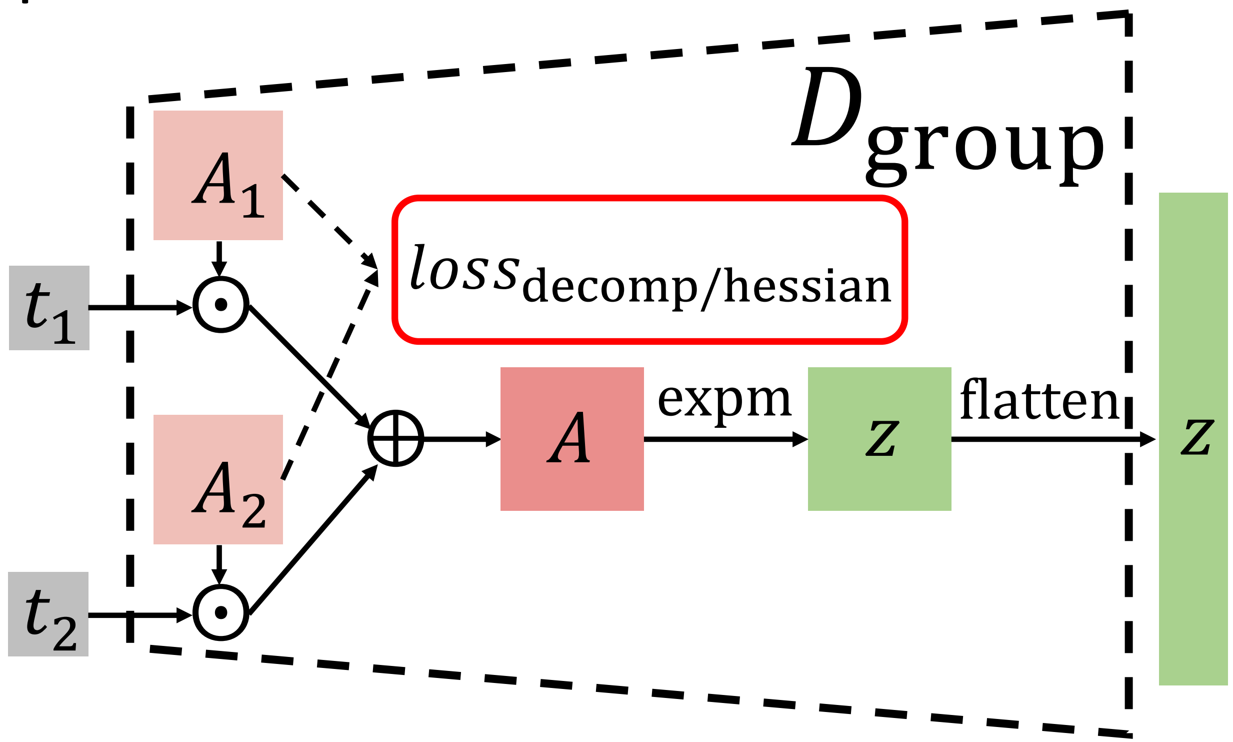
Method: (4) Constructing a VAE Model

Our proposed Lie Group VAE is a slight variant of bottleneck-VAE, which has a special group feature decoder:



Method: (4) Constructing a VAE Model

Inside the group decoder:



Note that the Lie algebra basis $\{A_i\}_{i=1}^m$ are learnable.

Prop. 1 and Prop. 2 are implemented as regularizations on $\{A_i\}_{i=1}^m$.
We refer a model with these constraints as Commutative Lie Group VAE.

Experiments

Ablation study on DSprites:

Models	FVM	SAP	MIG	DCI
VAE	$69.4_{\pm 10.9}$	$19.7_{\pm 10.6}$	$7.8_{\pm 6.4}$	$8.1_{\pm 4.1}$
+bottle	$74.6_{\pm 8.1}$	$29.2_{\pm 12.1}$	$12.9_{\pm 6.6}$	$11.6_{\pm 3.3}$
+exp	$\mathbf{83.6}_{\pm 3.2}$	$\mathbf{40.7}_{\pm 12.2}$	$\mathbf{17.2}_{\pm 6.8}$	$\mathbf{15.1}_{\pm 2.4}$

Table 1. Ablation study of bottleneck-VAE and exponential map on DSprites.

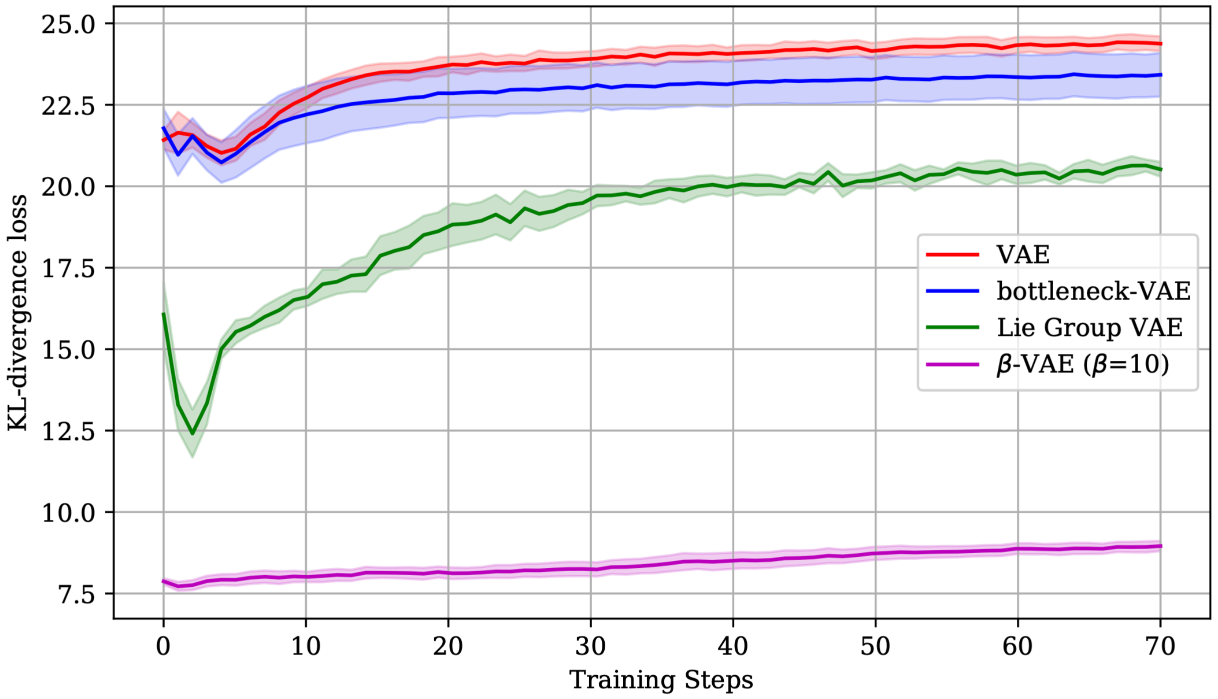


Figure 3. How the KL-divergence loss $(KL(q(t|x)||p(t)))$ evolves during training for different models.

Experiments

Ablation study on DSprites:

λ_{decomp}	FVM	SAP	MIG	DCI
0	$83.6_{\pm 3.2}$	$40.7_{\pm 12.2}$	$17.2_{\pm 6.8}$	$15.1_{\pm 2.4}$
5	$84.0_{\pm 3.9}$	$45.4_{\pm 11.5}$	$20.5_{\pm 6.9}$	$16.8_{\pm 4.3}$
20	$\mathbf{85.8}_{\pm 6.9}$	$48.7_{\pm 8.4}$	$23.6_{\pm 5.0}$	$18.2_{\pm 3.0}$
40	$85.5_{\pm 2.2}$	$\mathbf{50.8}_{\pm 5.0}$	$\mathbf{25.4}_{\pm 6.1}$	$\mathbf{19.7}_{\pm 4.6}$
80	$85.5_{\pm 4.8}$	$47.1_{\pm 8.6}$	$23.3_{\pm 6.2}$	$18.3_{\pm 6.5}$

Table 3. Ablation study of one-parameter decomposition on DSprites.

$\text{Size}_{\text{group}}$	FVM	SAP	MIG	DCI
4	$23.6_{\pm 3.3}$	$6.3_{\pm 6.0}$	$4.2_{\pm 3.9}$	$3_{\pm 0.5}$
9	$57.4_{\pm 5.8}$	$34.1_{\pm 12.9}$	$17.3_{\pm 7.4}$	$12.4_{\pm 4.4}$
25	$79.8_{\pm 2.8}$	$39.6_{\pm 13.4}$	$20.6_{\pm 8.5}$	$19.9_{\pm 3.8}$
64	$82.7_{\pm 3.7}$	$42.2_{\pm 12.5}$	$22.1_{\pm 10.1}$	$\mathbf{20.0}_{\pm 6.8}$
81	$84.4_{\pm 2.6}$	$45.2_{\pm 10.5}$	$23.0_{\pm 8.4}$	$19.6_{\pm 6.3}$
100	$\mathbf{85.5}_{\pm 2.2}$	$\mathbf{50.8}_{\pm 5.0}$	$\mathbf{25.4}_{\pm 6.1}$	$19.7_{\pm 4.6}$

Table 2. Ablation study of group size on DSprites.

λ_{hessian}	FVM	SAP	MIG	DCI
0	$83.6_{\pm 3.2}$	$40.7_{\pm 12.2}$	$17.2_{\pm 6.8}$	$15.1_{\pm 2.4}$
5	$83.8_{\pm 2.4}$	$46.8_{\pm 12.8}$	$19.8_{\pm 8.6}$	$17.5_{\pm 5.6}$
20	$86.1_{\pm 1.8}$	$\mathbf{54.1}_{\pm 1.2}$	$\mathbf{29.7}_{\pm 3.1}$	$\mathbf{23.4}_{\pm 4.1}$
40	$\mathbf{86.2}_{\pm 1.8}$	$48.2_{\pm 1.9}$	$25.2_{\pm 8.4}$	$19.1_{\pm 4.1}$
80	$85.0_{\pm 1.6}$	$43.6_{\pm 11.3}$	$20.1_{\pm 8.4}$	$17.4_{\pm 4.2}$

Table 4. Ablation study of Hessian penalty on DSprites.

Experiments

State-of-the-art comparison:

Model	DSprites	3DShapes
VAE	$69.4_{\pm 10.9}$	$83.6_{\pm 6.5}$
β-VAE	$74.4_{\pm 7.7}$	91 (Kim & Mnih, 2018)
Cascade-VAE	$81.74_{\pm 2.97}$	-
Factor-VAE	$82.15_{\pm 0.88}$	89 (Kim & Mnih, 2018)
Ours	$\mathbf{86.1}_{\pm 2.0}$	$\mathbf{93.2}_{\pm 4.0}$

Table 5. Unsupervised disentanglement state-of-the-art comparison on DSprites and 3DShapes.

Experiments

Qualitative results:

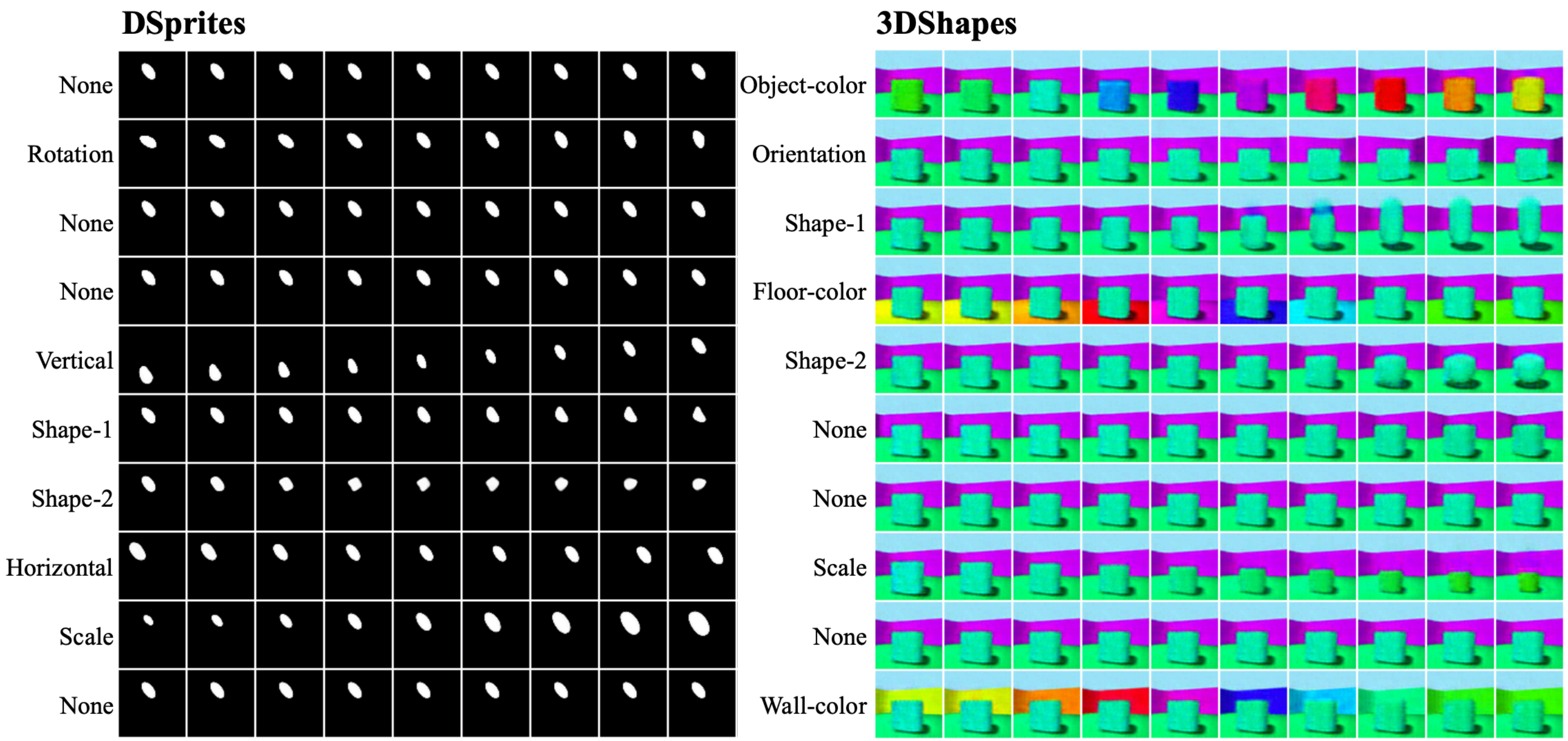
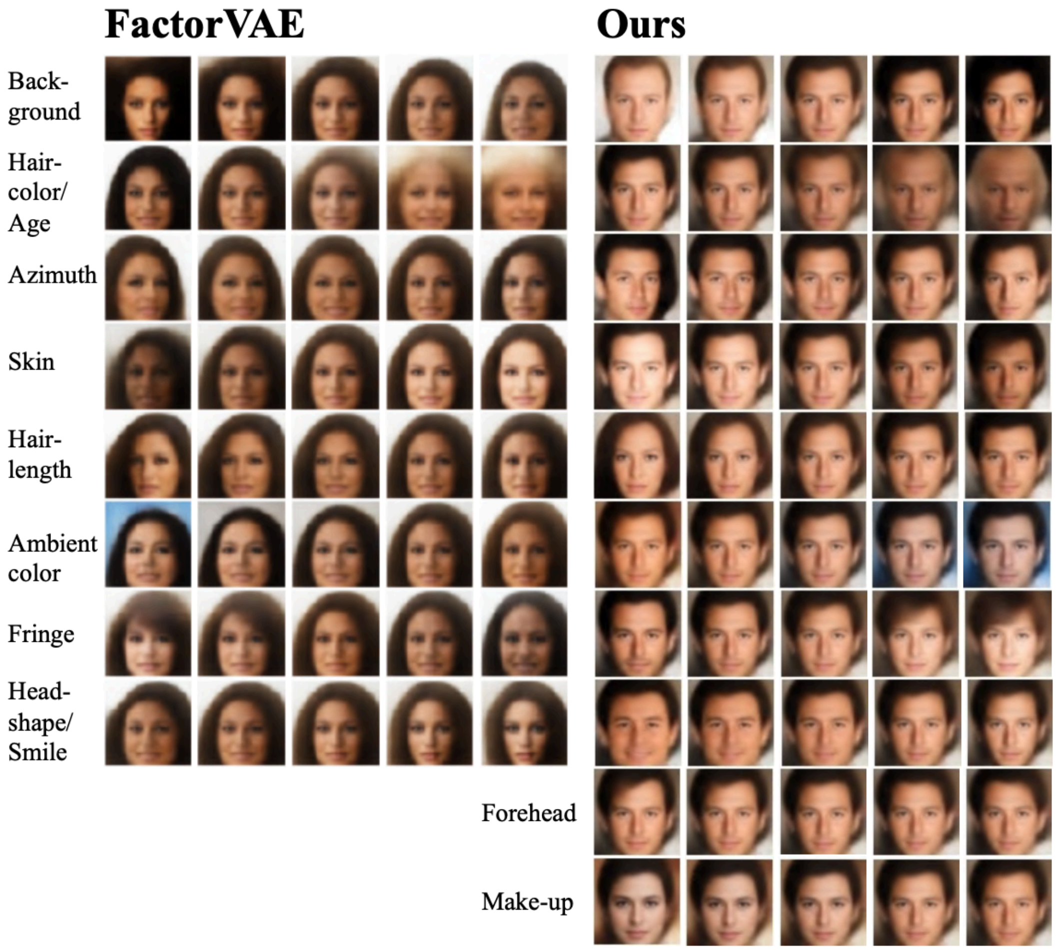
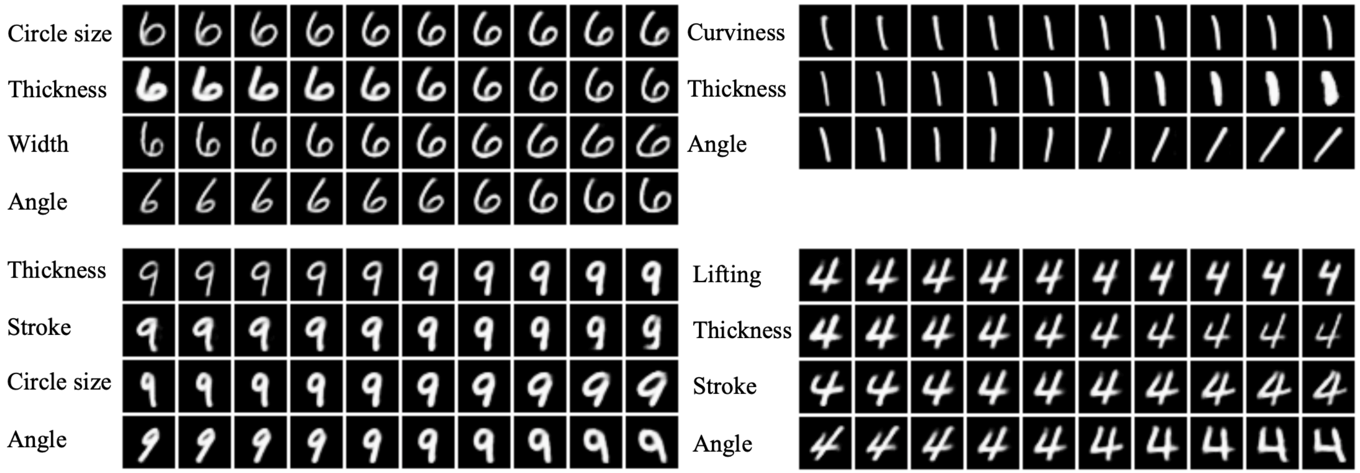


Figure 5. Latent traversals of our Commutative Lie Group VAE on DSprites and 3DShapes datasets.

Experiments

Qualitative results:





Thank you!