A Details of Shadow-Sunlight and Shadow-Pointlight

A.1 Controlled Setup in Blender

In this section, we describe the scene setting in Blender and how we set the generative values for each factor. The overview of the scene in Blender is shown in Fig. A.1, where the black box (⊠) is the camera. For Shadow-Pointlight, the light position is the value determined by the center of the light ball. For Shadow-Sunlight, since all rays are parallel, we set the direction from the center of light ball to the origin of scene to be the value of sunlight direction, the difference of two types of light sources in Shadow-Sunlight and Shadow-pointlight can be illustrated in Fig. A.2. For the scale, we set the basic size (Object scale=1), which is determined by the height from the center of object gravity to the floor. Other object scales are determined by changing this height.

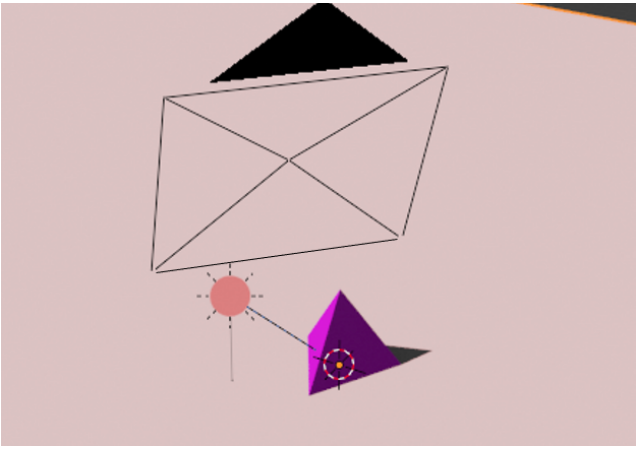

Figure A.1: Environment scene in Blender.

A.2 Factors of Variant

As introduced in Sec. 4, Shadow-Sunlight and Shadow-Pointlight have seven factors of variant and eight factors of variant, respectively. The name of each factor of variant and the number of each variant are summarized in Tab. A.1. Shadow datasets contain seven different object shapes: Cube, Sphere, Cylinder, Tetrahedron,140 Octahedron, Dodecahedron, and Icosahedron. Objects can have one of seven different colors in (R,G,B): Red (255,0,0), Orange (255,128,0), Yellow (255,255,0), Green (0,255,0), Cyan (0,255,255), Blue (0,0,255), and Purple (128,0,255). Each object can be assigned one of seven different scales. There are six different light colors (R,B,G): SkyBlue (128,255,255), Plum (255,128,255), Khaki (255,255,128), Lavender (128,128,255), Lightgreen (128,255,128), and Coral (255,128,128). In In Shadow-Sunlight, there are 20 different light directions, and in Shadow-Pointlight, the light directions are controlled by 20 different light positions. Since the shadow shape, the floor color, and the brightest floor position are effect factors, their values are controlled by object shape, object scale, light position/direction, and light color. To better illustrate each

Factors	Number of Variants
Object shape	7
Object color	7
Object scale	7
Light direction / position	20
Light color	6
Shadow shape	N/A
Floor color	N/A
Brightest floor position (only in pointlight)	N/A

Table A.1: Generative factor settings of Shadow-Sunlight and Shadow-Pointlight. Shadow shape, Floor color and Brightest floor position are effect factors controlled by cause factors.

Dataset	Resolution	Factors of Variation	Number of Samples	Nuisance	Various generative factors	Consistent with Causal Graph
Pendulum [CS]	96×96	4	7000	✗	✓	✓
Flow [CS]	96×96	4	7000	✗	✓	✓
CelebA(BEARD) [CS]	128×128	4	202599	✓	✗	✗
CelebA(SMILE) [CS]	128×128	4	202599	✓	✗	✗
Shadow-Sunlight	128×128	7	41160	✓	✓	✓
Shadow-Pointlight	128×128	8	41160	✓	✓	✓

Table A.2: Meta-data comparison between Shadow datasets and existing datasets.

generative factor of variation, we show the traversal of each generative factor of Shadow-Sunlight in Fig. A.3 and each generative factor of Shadow-Pointlight in Fig. A.4.

A.3 Split of training and test

For both Shadow-Sunlight and Shadow-Pointlight, we split the dataset by taking 90% samples for training, which are 36160 samples for training. The remaining 10% samples are used for testing, which are 4116 testing samples.

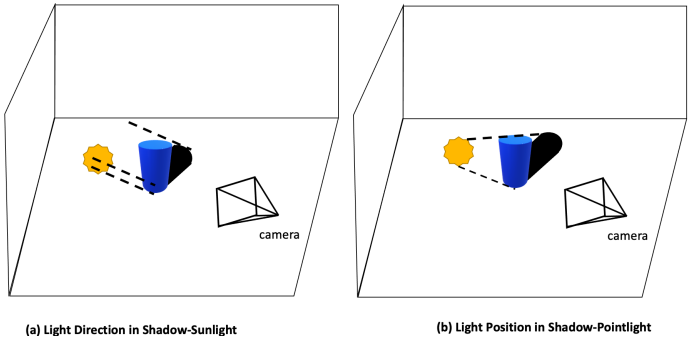

Figure A.2: Illustrations of two different light sources in Shadow-Sunlight and Shadow-Pointlight, respectively.

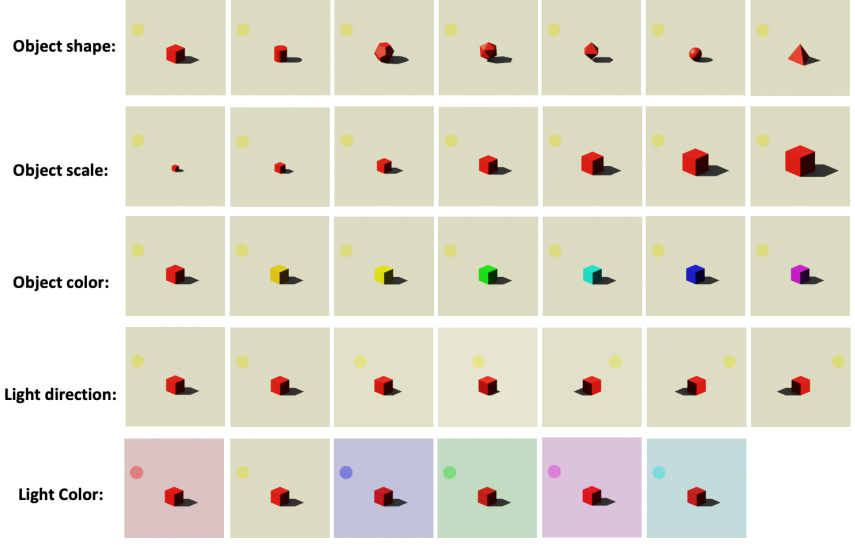

Figure A.3: Factors of variation in Shadow-Sunlight. There are 20 different values of Light direction factors.

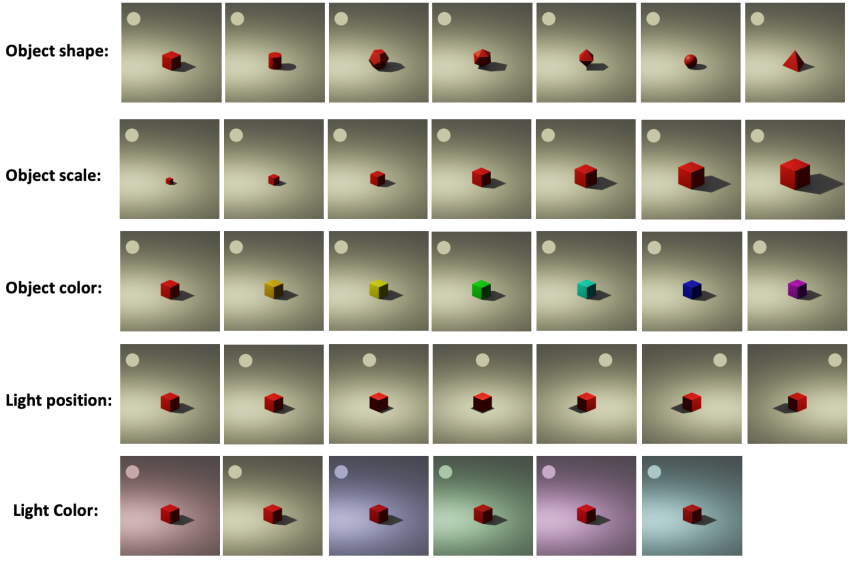

Figure A.4: Factors of variation in Shadow-Pointlight. There are 20 different values of Light position factors.

B Conditionally Independent Test on Original Real Datasets and Curate CelebA(BEARD)

As discussed in Sec. 4 and proposed by [28], the correctness of a proposed causal graph can be justified by testing the conditional independence between two factors. In causality theory, There are three types of graph building blocks: *chain*, *fork* and *immortality (collider)*. The structures of them are illustrated in Fig. A.5.

As shown in Fig. A.5(a) and (b) respectively, *Chain* and *fork* share the same set of dependencies. In both structures, x and X_3 are associated, i.e., they are mutually dependent on each other. If we condition on x', $(x|x')$ and $(X_3|x')$ become conditional independent with each other, i.e., $(x|x') \perp\!\!\!\perp (X_3|x')$.

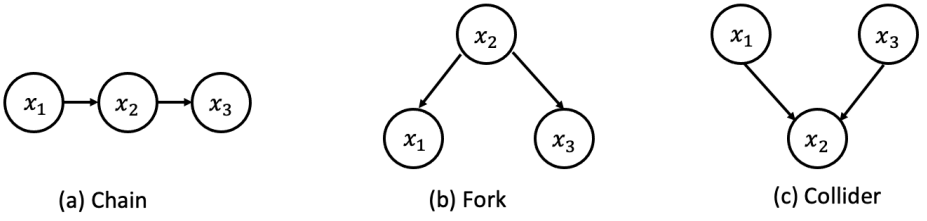

Figure A.5: Three types of causal graph building blocks.

Different from *chain* and *fork*, *Collider (immorality)* has a distinct set of dependencies, where the structure of *collider* is shown in Fig. A.5(c). x and X_3 are independent of each other but will become conditional dependent if both of them are conditioned on x', i.e., $(x|x') \not\perp\!\!\!\perp (X_3|x')$. Besides, when conditioned on descendants of x', $(x|de(x'))$ and $(X_3|de(x'))$ are also conditionally dependent, i.e., $(x|de(x')) \not\perp\!\!\!\perp (X_3|de(x'))$, where $de(x')$ stands for descendants of x'.

By utilizing the property of each basic block, we can determine the conditional independent relations between each factor according to the original proposed causal graphs of CelebA(BEARD) and CelebA(SMILE) [58], which are shown in Sec. 3.3(c) and (d). Although we discussed in Sec. 4 that we suggest omitting the evaluation on CelebA(SMILE), we also test the conditionally independent relations in CelebA(SMILE) and observe that data distribution of CelebA(SMILE) is also not consistent with the originally proposed causal graph. Discussed in Sec. 4, we incorporate χ^2 test to assess the conditional independent relations, where we set the significance level to be $\alpha = 0.05$. The null hypothesis is set as H_0: two factors are independent. Contrarily, H_1 is: two factors are not independent. If the p-value is less than the significance level $\alpha = 0.05$, we reject H_0. Since all generative factors in CelebA are binary, the degree of freedom is 1. We describe the test for CelebA(BEARD) and CelebA(SMILE) test in the following paragraphs sequentially.

CelebA(BEARD) conditionally independent tests The originally proposed causal graph of CelebA(BEARD) is shown in Sec. 3.3(c), where Bald and Beard are colliders of Age and Gender. By utilizing the conditionally independent relations inherent in three basic blocks in a causal graph, according to the originally proposed causal graph of CelebA(BEARD), the conditionally independent relations are summarised as: (1) Age and Gender should be mutually independent with each other, (2) Age and Gender become conditionally dependent when conditioned on Bald or Beard, (3) Bald and Beard are mutually dependent with each

	Age	Gender	Bald	Beard
Age		10^{-5}	0	0
Gender	10^{-5}		0	0
Bald	0	0		0
Beard	0	0	0	

Table A.3: p-value of χ^2 (freedom=1) test between each factor when no condition on original CelebA(BEARD). The red number indicates inconsistent relations between two factors with original proposed causal graph.

Conditioned on	p-value
Bald	0.0019
Beard	0.0
Bald and Beard	0.1519

Table A.4: p-value of χ^2 (freedom=1) test between Gender and Age when conditioned on different factors on original CelebA(BEARD).

other, (4) Bald and Beard are conditionally independent when conditioned on both Age and Beard, (5) Any other remaining relations are dependent relations. Results of χ^2 tests between two factors, when they are not conditioned on any other factors, are shown in Tab. A.3. Since the p-value between Age and Gender is smaller than the significance level $\alpha = 0.05$, which indicates that Age and Gender are not mutually independent of each other.

As shown in Tab. A.4, other conditionally independent tests demonstrate the consistency between the collider structure of the original proposed causal graph with data distribution.

CelebA(SMILE) conditionally independent tests Similar to the test in CelebA(SMILE), according to the originally proposed causal graph of CelebA(SMILE), which is shown in Sec. 3.3(d), the corresponding conditionally independent relations are: (1) Gender and Smile are mutually independent, (2) Gender and Mouth open are mutually independent,(3) Gender and Smile are conditionally dependent when conditioned on Eyes open, (4) Gender and Mouth open are conditionally dependent when conditioned on Eyes open, (5) any other remaining relations are dependent relations. Results of χ^2 tests between two factors when there is no condition are shown in Tab. A.5. When conditioned on Smile, according to the original proposed CelebA(SMILE) causal graph, the Gender and Mouth open should be independent, which is not aligned with the data distribution shown in Tab. A.6.

Curate CelebA(BEARD) and conditionally independent tests on curate CelebA(BEARD) As shown by the conditionally independent tests about the original CelebA(BEARD) dataset, the only inconsistent relation is the relation between Gender and Age, where they are sup-

	Gender	Smile	Eyes open	Mouth open
Gender		3×10^{-5}	0.02	0
Smile	3×10^{-5}		0	0
Eyes open	0.02	0		0
Mouth open	0	0	0	

Table A.5: p-value of χ^2 (freedom=1) test between each factor when no condition on original CelebA(SMILE). The red number indicates inconsistent relations between two factors with original proposed causal graph.

Conditioned on	p-value
Eyes open	0.0017
Smile	7×10^{-6}

Table A.6: p-value of χ^2 (freedom=1) test between Gender and Mouth open when conditioned on different factors on original CelebA(SMILE).

	Age	Gender	Bald	Beard
Age		0.5	0	0
Gender	0.5		0	0
Bald	0	0		0
Beard	0	0	0	

Table A.7: p-value of χ^2 (freedom=1) test between each factor when no condition on curated CelebA(BEARD).

posed to be mutually independent according to the causal graph. Thus, to address this issue, we explicitly sample the uniform Gender, where a random sample is first chosen, then according to the Gender and Age of that sample, we sample another random sample with the same Age but opposite gender. By performing this curate, in the curate CelebA(BEARD), we can make Gender and Age to be statistically independent of each other, which is consistent with the originally proposed causal graph.

By applying this curate operation, we explicitly make the Gender and Age to be mutually independent of each other. To justify the consistency between the data distribution in curated CelebA(BEARD) and the originally proposed causal graph, we conduct the same conditionally independent tests on curated CelebA(BEARD). As shown in Tab. A.7 and Tab. A.8, the data distributions in curated CelebA(BEARD) are aligned with the originally proposed causal graph. Thus, the curated CelebA(BEARD) can be used to evaluate causal representation learning.

Conditioned on	p-value
Bald	0.0
Beard	0.0
Bald and Beard	0.2459

Table A.8: p-value of χ^2 (freedom=1) test between Gender and Age when conditioned on different factors on curate CelebA(BEARD).

C Notation

As mentioned in Sec. 3, we include all notations and their corresponding meaning in this section, which can be shown in Tab. A.9.

Notation	Meaning
x and x'	input images pair
$\hat{x}'$ and $\hat{x}'$	reconstruction images pair
u and u'	micro-scope representation pair encoded from image pairing x and x', respectively
z and z'	causal factor pair mapped from u and u', respectively
$\tilde{z}$ and $\tilde{z}'$	New causal factor pair obtained from Effect Swap operation applied on z and z'
$\hat{z}$ and $\hat{z}'$	causal factor pair which is the outputs of GAE layer that takes $\tilde{z}$ and $\tilde{z}'$, respectively
$\hat{u}$ and $\hat{u}'$	micro-scope representation pair mapped from $\hat{z}$ and $\hat{z}'$, respectively
$\mathrm{Enc}(\cdot)$	Visual encoder, which takes image x as input and produce micro-scope presentation u as output
$\mathrm{Dec}(\cdot)$	Visual decoder, which takes micro-scope presentation u or $\hat{u}$ as input and produce reconstruction $\hat{x}$ as output
$f_{ma}(\cdot)$	Macro-embedding layer, which takes micro-scope presentation u as input and produce causal factor z as output
$\mathrm{GAE}(\cdot)$	Graph autoencoder, which takes causal factor z as input and produce causal factors $\hat{z}$ as output, which obey the causal mechanism
$f_{mi}(\cdot)$	Micro-embedding layer, which takes causal factors as input and produces micro-scope representation as output. It can be viewed as inverse function of $f_{ma}(\cdot)$

Table A.9: Notations for input images, micro-scope representations, causal representations, and modules.

D Identifiability Proof and Empirical evidence

D.1 Theoretical Identifiability Proof

In Sec. 3, we assume the generation process of causal representation learning datasets can be expressed as following Eqs. (9) to (11):

$$p(z) = \prod_i (z_i | pa_i), \qquad p(z') = \prod_i (z'_i | pa'_i) \tag{9}$$

$$z = s(z) \qquad z' = s(f(z, z', e)) \qquad e \sim p_E \tag{10}$$

$$x = g^*(z) \qquad x' = g^*((s(f(z, z', e)))), \tag{11}$$

where both z and z' are causal factors and obey the same causal mechanism, and are from the same distribution, pa_i is the set containing parents of factor i. E is the set of effect factor indices, and e is randomly extracted from such set according to probability p_E. f is an operation that involves replacing the value of one effect factor z'_e in z' with the value of paired effect factor z_e. s is the solution function that propagates causal relations from causes to their corresponding effects by $z_i = s_i(pa_i)$. s can be implemented as any neural network that obeys causality constraints [26, 40, 41]. Due to the constraints imposed by s, the effect factor value is exclusively determined by its parents. Thus, modifying the value of an effect factor does not affect the output of s. where pa_i is the set of parent factors of factor i and s is the solution function, which propagates the causal effects from causes to their corresponding effects. NOTEARS [41] defines s as a linear function with acyclicity constraint, which is utilized as a causal layer in CausalVAE [38]. For our proposed method, in order to handle the non-linearity of causal relations, we adopt graph autoencoder (GAE) [26] to propagate causal effects. In Eq. (10), f is the swap operation, where we replace one effect factor in z', e.g., z'_i with the effect factor in the same location of z. Consider the generative process in Eqs. (9) to (11) and further assume $p(z_i), p(z'_i)$ are continuous distribution, g^* is a smooth and invertible function, i.e., z and X which is the domain of x and x' are diffeomorphisic. Given unlabelled data x and x' and effect factor i, after swapping operation in Eq. (10). Then, the learned marginalized posterior $q(\hat{z})$ is a coordinate-wise reparameterization of the ground-truth $p(z)$ up to a permutation of indices.

The logic of the proof follows the proof in [24]. However, since Locatello et.al [24] assume the factors in latent representation are mutually independent, which is different from the CRL assumption, we adjust the generation processes and training procedure to make them align with the CRL assumption, i.e., the value of factors and raw inputs are generated through a static causal mechanism. In order to prove the identifiability, we follow these steps:

1. We first characterize the constraints that need to hold for the posterior $q(\hat{z}|x)$.

2. We parameterize all candidate posteriors $q(\hat{z}|x)$ as function g^*.

3. We show that for one causes-effect relation, i.e., one effect factor and its corresponding parent factors, $q(\hat{z}|x)$ disentangle such effect factor, its corresponding parent factors, and other factors.

4. We show that applying swapping operation on all causes-effect relations implies that every candidate posterior is a coordinate-wise reparameterization of the distribution of the ground-truth factors of variation.

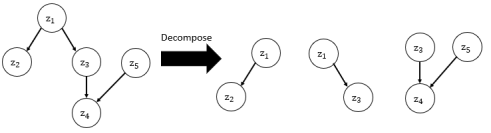

Figure A.6: Example of decomposing a causal graph into multiple causes-effect relations.

Step 1 By the generation process Eqs. (9) to (11), we know that all smooth and invertible functions g need to obey the following equations with probability 1, which is irrespective of whether $p(z)$ or $p(z')$ is used:

$$g^{-1}(x) = z = s(f(z', z, e)) \tag{12}$$

$$g^{-1}(x') = z' = s(f(z, z', e)) \tag{13}$$

We now focus on only one causes-effect relation, i.e., there is only one effect factor i that will be swapped between z and z'. Thus, given the assumption that the value of each element is different, we can rewrite the Eqs. (12) and (13) as follows, where z_o means other factors except the specific effect factor i and its corresponding parent factors pa_i:

$$g_i^{-1}(x) = z_i = s_i(pa_i, z_i', z_o) = s_i(pa_i, \cdot, \cdot) \tag{14}$$

$$g_i^{-1}(x') = z_i' = s_i(pa_i', z_i, z_o') = s_i(pa_i', \cdot, \cdot) \tag{15}$$

$$g_i^{-1}(x) = z_i \neq s_j(pa_j, z_j, z_{o_j}) = z_j \tag{16}$$

The reason we can focus on one causes-effect relation is that one effect is only decided by its corresponding parent factors, and a specific causal graph can always be decomposed into multiple causes-effect relations. One simple example can be shown in Fig. A.6.

Step 2 Per our assumption, X and z are diffeomorphic. Thus, all invertible smooth candidate functions for these two domains can be expressed as $g = g^* \circ h$, where h maps one point in $\hat{z}$ to one point in z, i.e., $h : \hat{z} \to z$, and h is also a smooth invertible function with inverse h^{-1} that maps z to $\hat{z}$. After we have $g^{-1} = h^{-1} \circ g*^{-1}$, we can express Eqs. (14) to (16) as:

$$\begin{aligned} h_i^{-1}(z) &= h_i^{-1}(s(f(z', z, e))) \\ &= h_i^{-1}(s_i(f(pa_i, z_i', z_o))) = h_i^{-1}(s_i(pa_i)) \end{aligned} \tag{17}$$

$$\begin{aligned} h_i^{-1}(z') &= h_i^{-1}(s(f(z, z', e))) \\ &= h_i^{-1}(s_i(f(pa_i', z_i, z_o))) = h_i^{-1}(s_i(pa_i')) \end{aligned} \tag{18}$$

$$h_i^{-1}(z) \neq h_j^{-1}(s(f(z, z', e))) = h_j^{-1}(s_j(pa_j)) \tag{19}$$

Because h is a smooth and invertible function, we know that h^{-1} maps the coordinate subspace pa_i to submanifold $\mathcal{M}_{pa}$ in $\hat{z}$ and coordinate subspace i to submanifold $\mathcal{M}_i$ in $\hat{z}$, and subspace o to submanifold $\mathcal{M}_o$, where each submanifold is disjoint from each other.

Step 3 Next, we shall see that for a fixed causes-effect relation, the only admissible functions $h : \hat{z} \to z$ are identifying three groups of factors pa_i, i, and o. To show this, we prove that h can only satisfy Eqs. (17) to (19) if it aligns the coordinate subspaces pa_i, i and o of z with the coordinate subspaces $\hat{pa}_i$, $\hat{i}$ and $\hat{o}$ of $\hat{z}$. We show this by contradiction. If i does not lie in $\mathcal{M}_{\hat{i}}$, then both Eqs. (17) and (18) will be violated. If pa_i does not lie in $\mathcal{M}_{\hat{pa}_i}$, then both Eqs. (17) and (18) will also be violated. Further, if o does not lie in $\mathcal{M}_{\hat{o}}$, then Eq. (19) will be violated. Therefore, Eqs. (17) to (19) can only be satisfied if h^{-1} maps each coordinate pa_i, i and o to a unique matching coordinate $\hat{pa}_i$, $\hat{i}$ and $\hat{o}$, where there exists a permutation π on $[d]$, where d is size of the latent space and $[d] = 1, 2, 3, ...d$, such that:

$$h^{-1}_{\hat{pa}_i}(z) = \tilde{h}_{\hat{pa}_i}(z_{\pi(pa_i)}) \tag{20}$$

$$h^{-1}_{\hat{i}}(z) = \tilde{h}_{\hat{i}}(z_{\pi(i)}) \tag{21}$$

$$h^{-1}_{\hat{o}}(z) = \tilde{h}_{\hat{o}}(z_{\pi(o)}) \tag{22}$$

This means that the jacobian of $\tilde{h}$ is block diagonal with blocks corresponding to coordinates indexed by $\hat{pa}_i$, $\hat{i}$ and $\hat{o}$.

Step 4 By step3, we show that for fixed causes-effect relations, we can find the permutation π and $\tilde{h}$ where the learned $\hat{z}$ is block-wise reparameterization corresponding to coordinates indexed by $\hat{pa}_i$, $\hat{i}$, and $\hat{o}$. However, the factors inside $\hat{pa}_i$ and $\hat{o}$ may still be unidentified. To finally achieve the coordinate-wise reparameterization, we now make the causes-effect relation randomly chosen. Since the cause-effect relation is randomly chosen, from step 3, we know that for one specific causes-effect relation, there is one $\tilde{h}$ function satisfying Eqs. (20) to (22). When a causes-effect relation is randomly chosen, Eqs. (20) to (22) are satisfied for all cases if and only if for every causes-effect relations, $\tilde{h}$ is block-diagonal. Thus, together with Eqs. (20) to (22), we have:

$$h^{-1}_i(z) = \tilde{h}_i(z_{\pi(i)}), \forall i \in [d] \tag{23}$$

π is a permutation on $[d]$, which implies that the jacobian of $\tilde{h}$ is diagonal. Therefore, we can express the marginalized posterior $q(\hat{z})$ as:

$$\begin{aligned} q(\hat{z}) &= p(\tilde{h}(z_{\pi([d])})) \mid det \frac{\partial}{\partial z_{\pi([d])}} \tilde{h} \mid \\ &= p(\tilde{h}(z_{\pi([d])})) \prod_i^d \mid \frac{\partial}{\partial z_{\pi([i])}} \tilde{h}_i \mid \end{aligned} \tag{24}$$

By further assuming $\mid \frac{\partial \tilde{h}_i}{\partial z_{\pi([i])}} \mid \neq 0, \forall i \in [d]$, Eq. (24) shows that $q(\hat{z})$ is coordinate-wise reparameterization of $p(z)$ up to a permutation of the indices.

D.2 Empirical identification evidence

As discussed in [23], unsupervised learning methods are empirically proven to be incapable of learning disentangled representation by two pieces of evidence. Firstly, different unsupervised learning methods have similar performance. Secondly, randomness is more important

than hyper-parameter tuning, where a model trained with sub-optimal hyper-parameters can even outperform the model trained with optimal hyper-parameters.

Following this empirical study, we examine the validity of our model from these two perspectives. Firstly, as shown in Secs. 4 and 5.1 and Fig. 3, all unsupervised learning methods have similarly poor performance while our model significantly outperforms them, which implies that the correct causal relations are learned by our model. Secondly, we provide an additional study regarding the effectiveness of hyper-parameter tuning. Our model is trained on Pendulum dataset with five different random hyper-parameters settings and 20 random seed for each hyper-parameters setting. As shown in Tabs. A.10 to A.13, compared to the optimal hyper-parameters setting, models trained with other hyper-parameters show worse performance. To this end, these two pieces of evidence entail the validity and soundness of the proposed model.

Models	PosMIC ↑	PosTIC ↑	NegMIC ↑	NegTIC ↓	F_1^{MIC} ↓	F_1^{TIC} ↑
original	56.1 ±2.1	43.6 ±3.7	29.7 ± 4.4	23.5 ±2.2	60.4 ±4.1	54.3 ±4.4
sub-opt 1	50.1 ±2.6	40.3 ±5.7	28.2 ± 4.2	23.9 ±2.7	55.3 ±3.9	49.1 ±4.2
sub-opt 2	46.6 ±2.4	39.7 ±3.0	24.3 ±2.4	20.5 ±2.3	49.2 ±3.2	43.2 ±3.3
sub-opt 3	49.7 ±3.2	40.7 ±2.1	28.9 ±2.4	23.7 ±4.0	55.1 ±3.4	51.8 ±3.9
sub-opt 4	52.5±2.8	42.2 ±3.3	29.9±3.9	24.0±3.6	58.6 ±4.0	52.2 ±3.6
sub-opt 5	47.6 ±5.1	40.6 ±3.6	26.6 ±2.6	22.3 ±2.0	52.9 ±3.6	47.7 ±4.1

Table A.10: Causal representation metrics tested on Pendulum where models trained with sub-optimal hyper-parameters are evaluated and each hyper-parameters model is trained under 20 different random seed.

Models	PosMIC ↑	PosTIC ↑	NegMIC ↑	NegTIC ↓	F_1^{MIC} ↓	F_1^{TIC} ↑
original	61.7 ±4.8	52.4 ±4.7	25.1 ±4.4	18.1 ±4.0	63.9 ±3.7	62.1 ±3.6
sub-opt 1	58.9 ±4.2	49.6 ±4.0	26.4 ±4.2	19.3 ±3.7	60.2 ±3.6	59.3 ±3.9
sub-opt 2	55.9 ±3.7	46.8 ±4.1	24.5 ±3.8	17.1 ±3.2	59.3 ±3.3	55.7 ±3.2
sub-opt 3	54.4 ±3.5	45.7 ±3.7	23.1 ±3.9	16.7±3.9	58.7 ±3.6	57.9 ±3.5
sub-opt 4	53.2 ±4.0	44.8 ±3.5	23.7 ±4.0	17.2±3.5	58.2 ±4.1	57.6 ±3.3
sub-opt 5	57.1 ±4.2	49.2 ± 3.6	25.2 ±3.6	18.4±3.9	59.9 ±3.1	58.9 ±3.2

Table A.11: Causal representation metrics tested on Flow where models trained with sub-optimal hyper-parameters are evaluated and each hyper-parameters model is trained under 20 different random seed.

Models	PosMIC ↑	PosTIC ↑	NegMIC ↑	NegTIC ↓	F_1^{MIC} ↓	F_1^{TIC} ↑
original	56.7 ±2.4	44.0 ±3.2	28.5 ± 4.5	22.5 ±3.2	60.9 ±4.2	55.0 ±4.1
sub-opt 1	52.2 ±3.2	41.2 ±3.3	28.2 ± 3.9	21.2 ±3.3	57.2 ±3.1	52.2 ±4.0
sub-opt 2	54.1 ±3.3	42.3 ±3.1	28.8 ± 3.8	23.2 ±3.7	59.3 ±4.0	53.5 ±3.7
sub-opt 3	50.7 ±3.7	39.4 ±2.9	25.3 ± 3.5	20.7 ±2.9	55.3 ±3.7	50.3 ±3.6
sub-opt 4	52.1 ±2.9	40.9 ±3.2	26.8 ± 3.7	21.3 ±3.7	57.7 ±3.9	52.7 ±4.2
sub-opt 5	53.4 ±3.0	42.0 ±3.4	27.7 ± 4.0	22.4 ±2.9	58.2 ±4.1	53.3 ±4.3

Table A.12: Causal representation metrics tested on Shadow-Sunlight where models trained with sub-optimal hyper-parameters are evaluated and each hyper-parameters model is trained under 20 different random seed.

Models	PosMIC ↑	PosTIC ↑	NegMIC ↑	NegTIC ↓	F_1^{MIC} ↓	F_1^{TIC} ↑
original	59.2 ±4.9	44.3 ±4.7	34.1 ±3.7	28.3 ±3.2	60.3 ±3.7	53.3 ±4.2
sub-opt 1	56.3 ±4.2	43.0 ±3.9	38.1 ±3.3	28.2 ±2.7	57.3 ±3.6	51.2 ±4.1
sub-opt 2	54.7 ±4.3	42.2 ±4.2	35.2 ±3.5	27.5 ±2.9	56.7 ±4.2	49.7 ±3.9
sub-opt 3	53.2 ±3.9	40.7 ±3.5	33.7 ±4.2	26.1 ±3.5	55.1 ±3.9	49.3 ±3.7
sub-opt 4	54.1 ±4.0	42.4 ±3.3	34.9 ±3.3	28.6 ±3.1	55.9 ±3.2	50.7 ±4.4
sub-opt 5	57.3 ±4.4	43.6 ±3.7	36.7 ±3.9	27.7 ±3.8	59.0 ±4.0	52.5 ±4.1

Table A.13: Causal representation metrics tested on Shadow-Pointlight where models trained with sub-optimal hyper-parameters are evaluated and each hyper-parameters model is trained under 20 different random seed.

D.3 Limitation and future work

As discussed in Appx. D.1, the learned marginalized posterior $q(\hat{z})$ can be identified when we know the true parent factors of specific effect factors and the number of generative causal factors d. For practicality, we relax the first requirement by utilizing an HSIC regression module and graph autoencoder (GAE). However, we can not relax the second constraint, which requires prior knowledge about the ground-truth number of generative causal factors. Another implicit assumption is causal sufficiency, which can also be assumed in all previous works of causal representation learning [3, 38, 42]. Causal sufficiency assumes that all factors are required to be observed during training. To mitigate the first challenge, methods of constraining the number of latent factors need to be adapted, for example, utilizing the intrinsic dimension estimation method [1, 19]. To address the second challenge, where there are some confounders that are not observed, a new type of causal graph instead of pure directed causal graph (DAG) needs to be adopted, such as maximal ancestral graph (MAG) [29]. In addition, modifications to current causal representation learning datasets are also needed to create a proper training and evaluation protocol to compare a model that can handle the hidden confounders with other models with no such ability.

Model	Pendulum		Flow		Shadow-Sunlight		Shadow-Pointlight	
	MIC ↑	TIC↑	MIC↑	TIC↑	MIC ↑	TIC↑	MIC↑	TIC↑
Fully supervised learning methods (all labels are used)								
CausalVAE []	95.1 ±2.1	81.6±1.9	72.1±1.3	56.4±1.6	72.7±4.0	62.2±3.9	71.1±5.1	58.9±4.3
ConditionVAE []	93.8±3.3	80.5±1.4	75.5±2.3	56.5±1.8	72.4±4.3	62.4±4.2	71.9±5.3	59.1±3.3
Unsupervised learning methods (no label is used)								
CausalVAE(unsup) []	21.2±1.4	12.0±1.0	20.5±4.7	11.8±2.6	19.3±4.4	10.0±3.5	17.7±4.3	8.7±2.8
β-VAE []	22.6±4.6	12.5±2.2	23.6±3.2	12.5±0.6	18.9±4.1	7.8±3.7	17.9±4.5	9.2±1.8
LadderVAE []	22.4±3.1	12.8±1.2	34.3±4.3	24.4±1.5	15.6±4.9	7.9±2.8	15.3±3.9	7.4±2.1
Weakly supervised learning methods (no label is used)								
DoVAE []	86.6±7.9	74.5±5.1	65.5±6.6	56.7±4.9	53.6±6.6	39.7±5.5	56.5±5.6	43.5±4.1
Causal-Macro(ours)	91.3±3.5	80.0±4.1	68.1±4.1	57.7±4.3	65.9±3.2	56.0±4.2	62.3±3.8	52.6±3.1

Table A.14: MIC and TIC values tested on Pendulum, Flow, Shadow-Sunlight and Shadow-Pointlight.

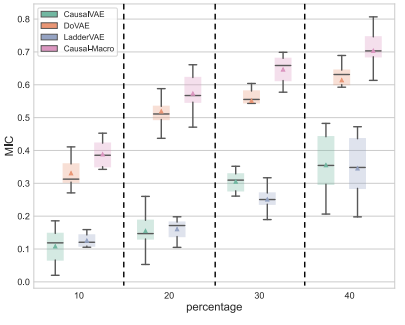

Figure A.7: MIC ↑ results on the curated CelebA(BEARD).

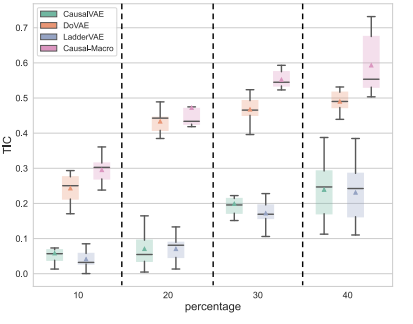

Figure A.8: TIC ↑ results on the curated CelebA(BEARD).

encoder	decoder
4*96*96*900 fc. 1ELU	concepts*(4*300 fc. 1ELU)
900*300 fc. 1ELU	concepts*(300*300 fc. 1ELU)
300*2*concepts*k fc.	concepts*(300*1024 fc. 1ELU)
-	concepts*(1024*4*96*96 fc.)

Table A.15: Pendulum and Flow datasets model architecture.

encoder	decoder
-	1*1 conv. 128 1LReLU(0.2), stride 1
4*4 conv. 32 1LReLU (0.2), stride 2	4*4 convtranspose. 64 1LReLU(0.2), stride 1
4*4 conv. 64 1LReLU (0.2), stride 2	4*4 convtranspose. 64 1LReLU(0.2), stride 1
4*4 conv. 64 1LReLU (0.2), stride 2	4*4 convtranspose. 32 1LReLU(0.2), stride 1
4*4 conv. 64 1LReLU (0.2), stride 2	4*4 convtranspose. 32 1LReLU(0.2), stride 1
4*4 conv. 256 1LReLU (0.2), stride 2	4*4 convtranspose. 32 1LReLU(0.2), stride 1
1*1 conv. 3, stride1	4*4 convtranspose. 3, stride 2

Table A.16: Shadow datasets and curated CelebA(BEARD) model architecture.

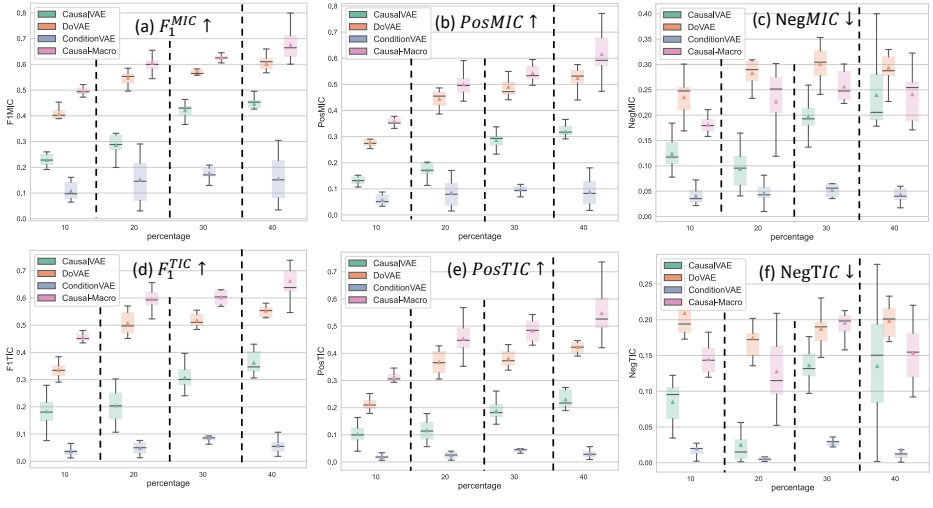

Figure A.9: Tested on the curated CelebA(BEARD), our method consistently outperforms SOTAs.

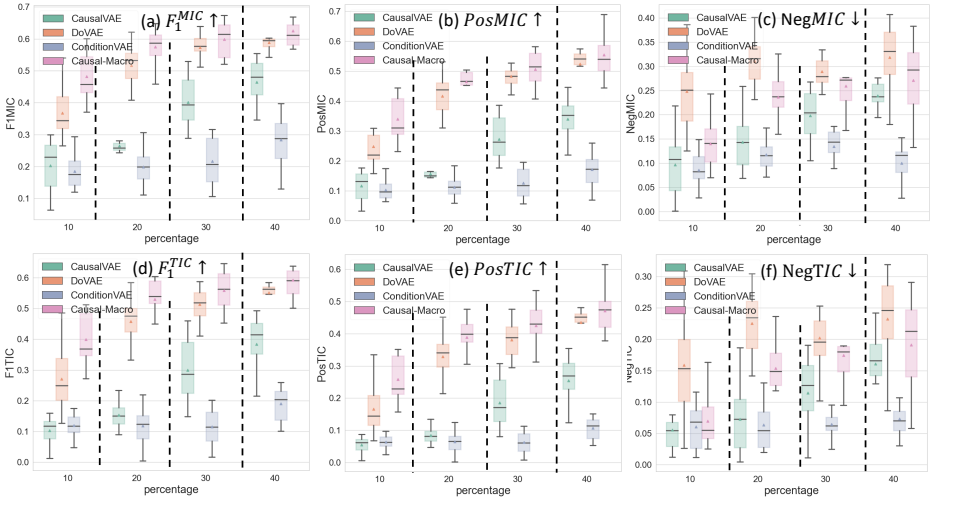

Figure A.10: Tested on the curated CelebA(SMILE), our method consistently outperforms SOTAs.

E Other experiments results

E.1 MIC and TIC scores tested on all datasets

As mentioned in Sec. 5.1, MIC and TIC do not evaluate causal relations, thus they are not metrics for evaluating causal representation learning [42]. However, since MIC and TIC estimate the mutual information between one latent element with its corresponding generative factor, MIC and TIC can reflect the performance of semantic meaning learning. To make more comprehensive comparisons among all models, we also include the results of MIC and TIC values of all models tested on Pendulum, Flow, Shadow-Sunlight, and Shadow-Pointlight in Tab. A.14. As shown by Tab. A.14, supervised learning methods, CausalVAE [38] and ConditionVAE [33], achieve best performance on MIC and TIC because of fully supervised learning. However, when no label is available, CausalVAE(unsup) [38] and other unsupervised learning methods fail to encode good semantic information in latent representation, where a supervision signal is required to achieve such goal [23]. Compared to unsupervised learning methods, weakly supervised learning methods can still successfully encode meaningful semantic information in latent representation because a supervision signal is introduced by using a pair of inputs. Compared with DoVAE, our Causal-Macro can achieve better performance on MIC and TIC because it first encodes visual information into micro-scope representation, which can decrease the difficulty of downstream tasks. DoVAE seeks to achieve both semantic information encoding and causal relations discovery simultaneously, which jeopardizes its performance on both tasks. As discussed in Secs. 4 and 5.1, ground-truth labels are required to force the model to focus on the four expected generative factors, thus all models are semi-supervised trained when tested on curate CelebA(BEARD). Our proposed method, Causal-Macro, consistently outperforms other methods under the same supervision strength, which indicates that Causal-Macro is able to better utilize the supervision signal introduced by a pair of inputs.

E.2 Total evaluation results on curated CelebA(BEARD) and CelebA(SMILE)

As discussed in Sec. 5 and shown in Fig. 3, because of the page limitation, we only include the F_1^{MIC} and F_1^{TIC} which are combinational metrics in main pages. In this section, we include all other metris results, including PosMIC/TIC and NegMIC/TIC of all models on curated CelebA(BEARD) and CelebA(SMILE). The results of curated CelebA(BEARD) and CelebA(SMILE) are shown in Fig. A.9 and Fig. A.10, separately.

E.3 Discussion about Causal-Macro and ILCM

In Sec. 5, our method is compared with ILCM [3] across multiple datasets: Pendulum, Flow, and our newly proposed Shadows datasets. Even though we have briefly described the difference between ILCM and our method in Sec. 3, in this section, we will delve into the distinctions between ILCM and our approach from various angles in more detail. These include differences in types of input pairs, applicable fields, and performance metrics across diverse applications. This comprehensive comparison will highlight the unique aspects and advantages of our proposed method in relation to ILCM.

Although both the proposed method and ILCM utilize pairs of inputs for learning causal representations, the specific pairs required by each method differ. This distinction high-

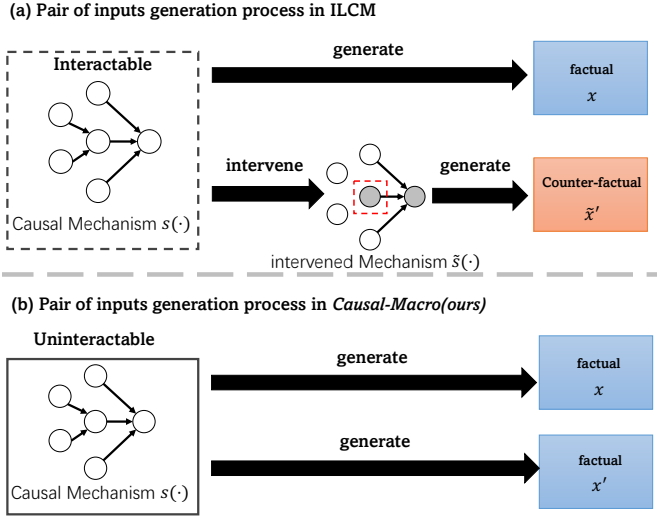

Figure A.11: Comparison between the different processes of generating a pair of inputs in WSCRL [3] and our work. Specifically, in WSCRL, to generate a pair of inputs, one sample is generated by the original causal system, and another sample in the pair is counterfactual, which is generated by the intervened causal system. Because of the requirement of applying intervention, WSCRL requires that the original causal system can be interacted with. On the contrary, since the proposed *Causal-Macro* takes both factual inputs, it is more suitable for cases where the causal system can not be interacted with and this is the focus of our work.

lights a key variation in their approaches to causal representation learning. In ILCM, as discussed in [3], two types of inputs are used to learn causal representation. The first is a natural observational sample produced by the original causal mechanism. The second is a sample generated after applying a perfect atomic intervention to this mechanism, where perfect atomic intervention means intervening on only one factor without altering other factors. Thus, in this process, two distinct causal mechanisms are employed: the original (producing a factual sample) and the modified (resulting in a counter-factual sample because the causal mechanism is modified, which is defined by [28]). In contrast, our method uses pairs of inputs that are both derived from the original causal mechanism without any intervention, making both samples factual. Such difference is also illustrated in Fig. A.11.

The input pair requirements for ILCM and our method lead to different applicable fields. ILCM, as described in [3], is ideal for scenarios where an agent can *interact with every node individually in a causal mechanism, allowing for interventions.* This makes it particularly effective in controlled experimental environments. In contrast, *Causal-Macro* takes pairs of inputs both derived from the unaltered causal mechanism, making it more suitable for situations where interaction with the causal mechanism is not possible and only passive observation is available. As demonstrated in our results, Sec. 4, *Causal-Macro* outperforms ILCM in scenarios where the causal mechanism remains un-intervened, and the inputs are factual.

It is important to note that when interventions on the causal mechanism are possible, ILCM exhibits superior performance. For this comparison, we used the CausalCircuit dataset from [3], also used in ILCM. CausalCircuit contains four (4) ground-truth generative factors,

and the ground-truth underlying causal mechanism simulates the control flow between lights in different colors. To adapt Causal-Macro to this setup, we trained it only with samples from the original causal mechanism, excluding those from the modified system. The results, detailed in Table Tab. A.17, show that ILCM outperforms our method when perfect interventions are feasible. This performance difference arises mainly from two factors: first, the reliance of our method on factual input pairs reduces the sample size compared to ILCM; second, the ability of ILCM to detect intervened nodes provides a stronger supervisory signal for learning causal representations.

In conclusion, the distinct input requirements of the proposed *Causal-Macro* and ILCM lead to varying suitable application scenarios. Our paper focuses on exploring Causal Representation Learning (CRL) in situations where interaction with the original causal mechanism is not possible. In addition, there are real-world scenarios where partial interaction with the causal mechanism is feasible, presenting new challenges for current CRL methods. We leave this area as an opportunity for future research.

Models	Causal-Circuits					
	PosMIC ↑	PosTIC ↑	NegMIC ↓	NegTIC ↓	F_1^{MIC} ↑	F_1^{TIC} ↑
ILCM [9]	79.4	69.3	24.1	17.6	79.7	74.6
Causal-Macro	70.6	64.4	33.0	25.7	68.8	65.5

Table A.17: Causal representation metrics tested on CausalCircuit.

F Implementation detail

F.1 Computation resources and model architecture

As our training and inference device, we employ a single NVIDIA 1080 Ti GPU. Following the design of CausalVAE [58], we exhibit the VAE architecture of synthetic datasets in Tab. A.15 and the VAE architecture of Shadow datasets and curated CelebA(BEARD) in Tab. A.16. We also use the CausalVAE configuration for latent representation, where latent space z is expanded to a matrix $z \in R^{n\times k}$, where n is the number of concepts and k is the latent dimension of each concept. k is set to 4 for VAE in the synthetic and Shadow datasets, and 32 for VAE in the CelebA dataset.

F.2 Effect Swap and RCDM

As described in Sec. 3, we incorporate a mask vector m, where the value of the chosen effect factor is one and the values of all other factors are zero. As discussed in Sec. 3, because we incorporate a graph autoencoder (GAE) [26] as solution function and all causal effects relations in GAE are propagated in one step, the mask m is necessary to remove the influence of the chosen effect factor to its possible children factors, which will make GAE output counter-factual result if such effect factor has children factor. We need to point out that such mask m is not necessary if the solution function is implemented as any topological method, i.e., the causal effects are propagated from root causes to the effect sequentially. However, the swap operation f is valid when implementing on root causes if the solution function is topological, and any incorrect causal relation will sequentially affect all relations in its descendants because the causal effects are propagated sequentially. As described in Sec. 3, our loss function for weakly supervised training is shown in Eq. (6), and the loss for

semi-supervised training is shown in Eq. (7). The hyperparameters (α, β) are grid search among $\{1e^{-3}, 1e^{-2}, 1e^{-1}, 1.0\}$. Further, as described in Sec. 3, we incorporate a Root Cause Discovery Module to identify the root causes in the causal graph. An MSE regularization is further added to a pair of latent elements at the same location when their D_{KL} is small to further encourage the self-dependency constraints on discovered root cause factors.

F.3 HSIC algorithm

Algorithm 1 HSIC regression detector

Input: maro-level representations $z = [z_1, z_2, ..., z_n]$
  $i := 0$, $j := 0$, $L := \mathbf{0}$
  while $i \leq n$ **do**
    while $j < i$ **do**
      $c := z_i$, $e := z_j$
      $\hat{e} := \Phi(c)$, $\hat{c} := \Psi(e)$
      $L_{ij} := HSIC(e, \hat{e})$
      $L_{ji} := HSIC(c, \hat{c})$
      $j := j + 1$
    end while
    $i := i + 1$
  end while
  $A = 1 - \sigma(L - max(L))$ ▷ σ is the sigmoid function
  return A ▷ A is causal graph matrix

As described in Sec. 3.3, given two causal associated factors X and Y, the direction with a smaller regression loss indicates higher confidence that this direction is the correct causal direction, i.e.,

$$L(Y, \Phi(X)) < L(X, \Psi(Y)) \Rightarrow X \text{ is cause of } Y, \tag{25}$$

When training, the causal discovery on macro-scope representation based on HSIC regression is described in Algorithm 1, where the causal direction between a pair of factors is indicated by comparing the element in its diagonal symmetric location.

G Traversal Visualization

We include the image reconstruction traversal results in this section which are shown in Figs. A.12 to A.15. When changing the latent cause factors before the causal discovery layer, the latent effect factors in the reconstruction are changed accordingly because the causal discovery layer propagates causal effects from causes to their corresponding effects. In contrast, when changing the latent effect factors after the causal discovery layer, since there are no causal effects propagated from the causes to effects after the causal discovery layer, the reconstructions can be counterfactual images, where the latent cause factors stay unchanged.

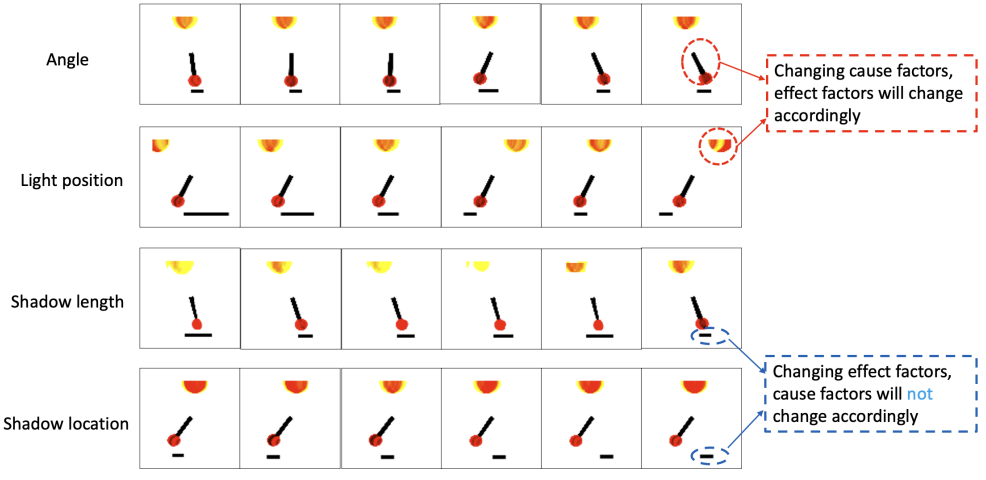

Figure A.12: Traversal reconstruction of Pendulum dataset. For each row, we only change one latent factor value and fix all other latent factors.

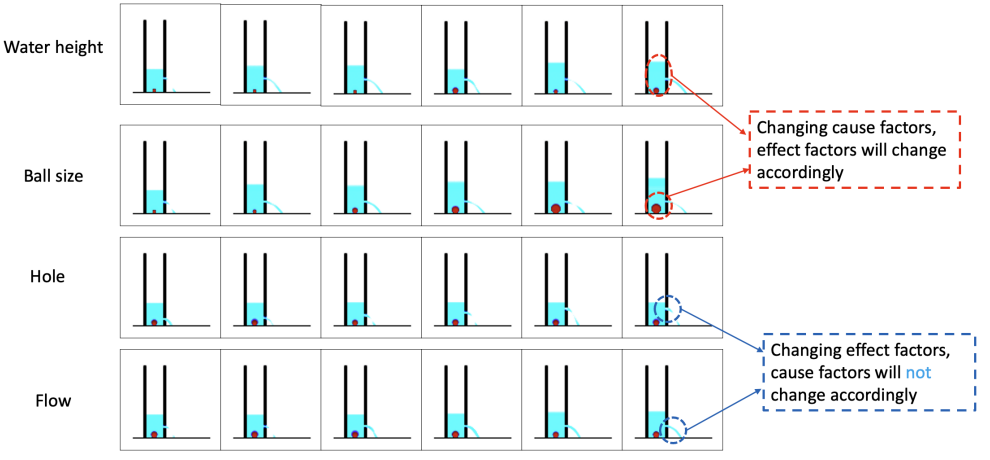

Figure A.13: Traversal reconstruction of Flow dataset. For each row, we only change one latent factor value and fix all other latent factors.

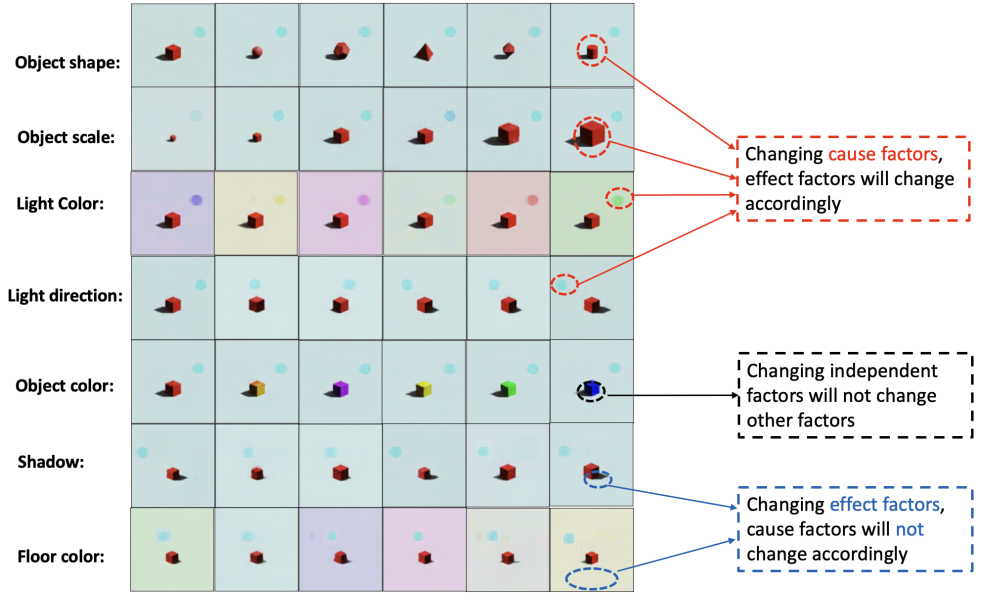

Figure A.14: Traversal reconstruction of Shadow-Sunlight dataset. For each row, we only change one latent factor value and fix all other latent factors.

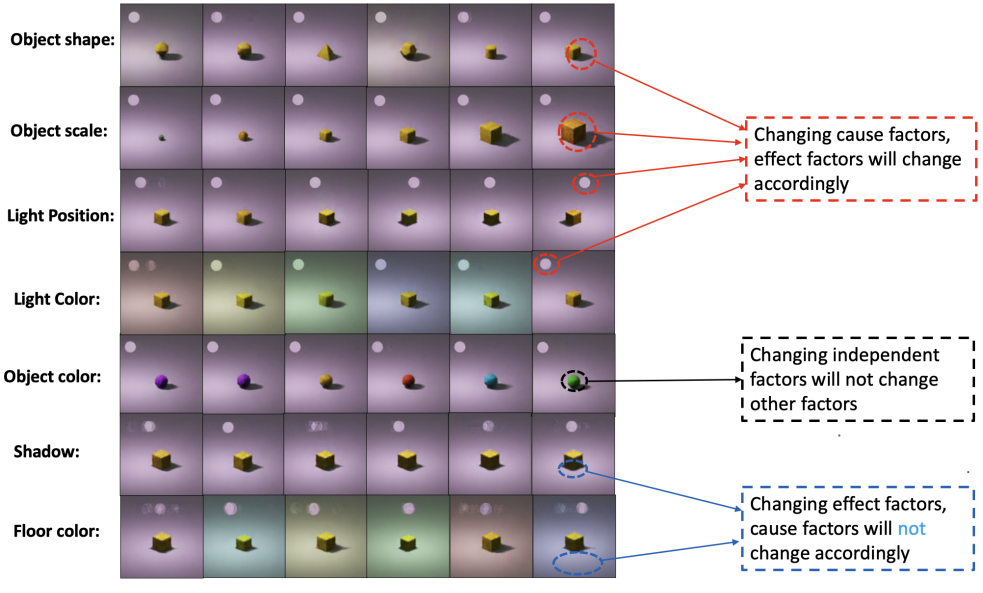

Figure A.15: Traversal reconstruction of Shadow-Pointlight dataset. For each row, we only change one latent factor value and fix all other latent factors.