Orthogonal Jacobian Regularization for Unsupervised Disentanglement in Image Generation Supplementary Material

Yuxiang Wei[1,*], Yupeng Shi[1], Xiao Liu[2], Zhilong Ji[2], Yuan Gao[2], Zhongqin Wu[2], Wangmeng Zuo[1,3] (✉)
[1]Harbin Institute of Technology, [2]Tomorrow Advancing Life, [3]Pazhou Lab, Guangzhou
{yuxiang.wei.cs, csypshi}@gmail.com {liuxiao15, jizhilong, gaoyuan23, wuzhongqin}@tal.com
wmzuo@hit.edu.cn

A. Proof of Proposition

We first give a brief proof of the proposition that related to SeFa [2] in the main paper.

Proposition 1. *Let* $\mathbf{W} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$ *be the singular value decomposition (SVD) of the weight parameter* $\mathbf{W}$. *Let* $\mathbf{z}' = \mathbf{V}^T\mathbf{z}$, $\mathbf{W}' = \mathbf{U}\mathbf{\Lambda}$, *and define* $G_1(\mathbf{z}) = \mathbf{W}\mathbf{z}$, $G_1'(\mathbf{z}') = \mathbf{W}'\mathbf{z}'$. *We have,*

1. $G_1'(\mathbf{z}')$ *is equivalent with* $G_1(\mathbf{z})$, *i.e.,* $G_1(\mathbf{z}) = G_1'(\mathbf{z}')$.
2. *Hard orthogonal Jacobian constraint can be attained, i.e.,*

$$\left[\frac{\partial G_1'}{\partial z_i'}\right]^T \frac{\partial G_1'}{\partial z_j'} = 0. \tag{1}$$

Proof.
1.

$$\begin{aligned} G_1(\mathbf{z}) &= \mathbf{W}\mathbf{z} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T\mathbf{z} = \mathbf{U}\mathbf{\Lambda}\mathbf{z}' \\ &= \mathbf{W}'\mathbf{z}' = G_1'(\mathbf{z}'). \end{aligned} \tag{2}$$

2.

$$\begin{aligned} \left[\frac{\partial G_1'}{\partial \mathbf{z}'}\right]^T \frac{\partial G_1'}{\partial \mathbf{z}'} &= [\mathbf{W}']^T\mathbf{W}' = \mathbf{\Lambda}^T\mathbf{U}^T\mathbf{U}\mathbf{\Lambda} \\ &= \mathbf{\Lambda}^2, \end{aligned} \tag{3}$$

where $\mathbf{\Lambda}$ is diagonal. $\left[\frac{\partial G_1'}{\partial z_i'}\right]^T \frac{\partial G_1'}{\partial z_j'}$ is the off-diagonal entry of Eqn. 3 when $i \neq j$, and thus to be zero. □

B. Additional Qualitative Results

B.1 CLEVR-1FOV Dataset

Fig. 1 shows the qualitative results on the CLEVR-1FOV by our OroJaR. CLEVR-1FOV has only one factor of variation: a red cube's location along a single axis. From Fig. 1, our OroJaR can successfully deactivate the redundant dimensions while controlling the position of the object with the only activated dimension (the top left row).

B.2 CLEVR-U Dataset

Fig. 2 shows the qualitative comparison with SeFa [2] and Hessian Penalty [1] on the CLEVR-U dataset. CLEVR-U indicates that we train the model on CLEVR-Simple (4 factors of variation, *i.e.*, horizontal and vertical positions, shape, and color) by setting the dimension of input to 3, which is an underparameterized setting. Obviously, SeFa fails to disentangle the four factors, which is shown that each dimension controls all four variations at the same time. Hessian Penalty also entangles the position with shape variation (*e.g.*, 2nd and 3rd rows). On the contrary, our OroJaR still learned to control the variations of horizontal position, vertical position, and shape (entangles with color) independently. The results indicate that our OroJaR is superior in disentangling spatially correlated variations (*e.g.*, shape and position).

B.3 BigGAN

Fig. 3 shows a more comprehensive qualitative comparison with Hessian Penalty [1] and Voynov [3] on BigGAN. One can see that, three methods discover similar latent directions (*i.e.*, zoom, rotate, and smoosh nose). However, Voynov [3] gives a degraded results when rotating the dog to the left side. It is worth noting that, Hessian Penalty finds directions that are very similar to ours. For rotation and smoosh nose directions, the cosine similarities between our method and Hessian Penalty reach 0.99. This may be caused by the limitation of pre-trained GAN. Nonetheless, compared with Hessian Penalty, our method performs a better zoom editing quality.

*This work was done when Yuxiang Wei was a research intern at TAL

Table 1: Comparison of Variation Predictability Metric (VP) for different settings and SeFa on Dsprites.

Method	GAN	L0	L1	L2	L3	L4	L0∼1	L0∼2	L0∼3	L0∼4	SeFa	Ours(L0∼3)
VP(%, ↑)	30.9	47.6	48.1	50.2	36.4	35.0	48.8	53.5	**54.7**	52.3	48.6	**54.7**

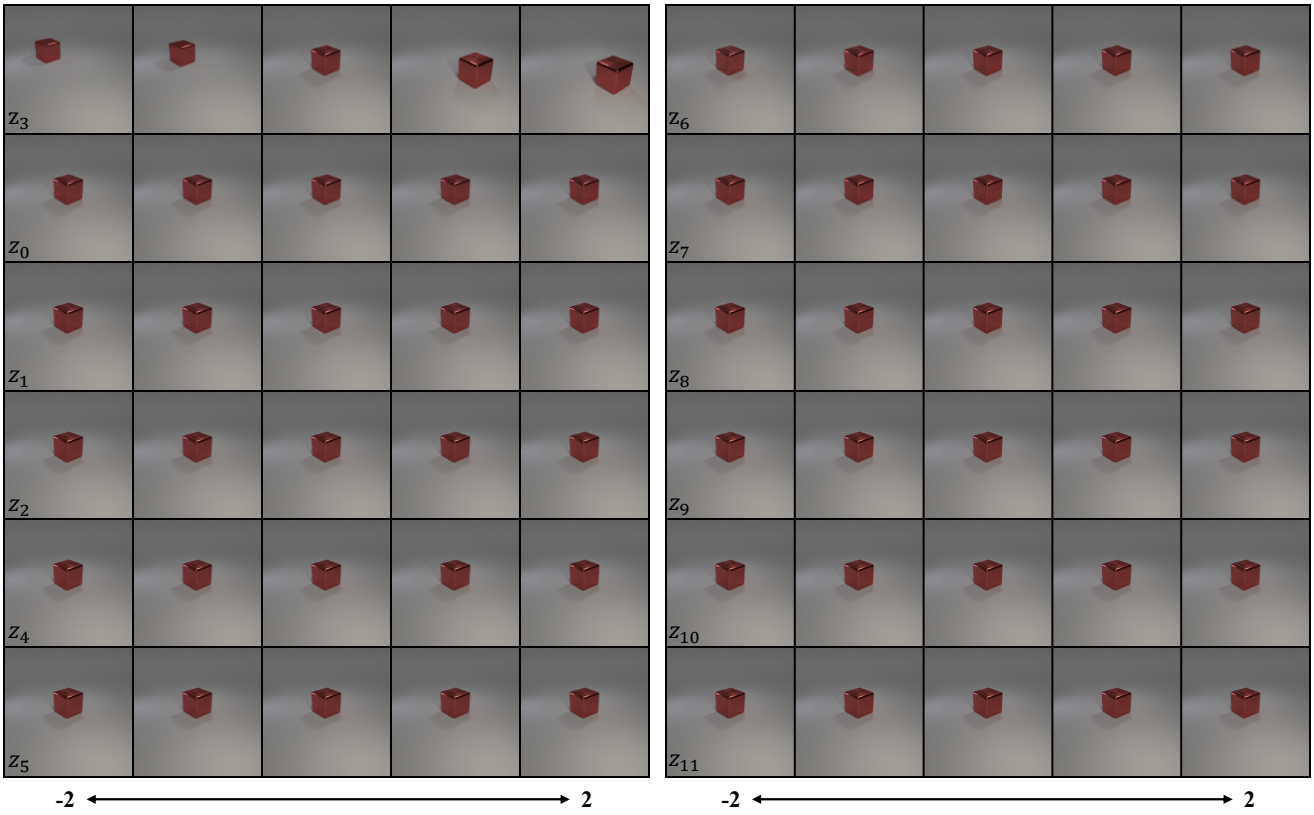

Figure 1: Qualitative results on the CLEVR-1FOV by our OroJaR. We randomly sample a 12-dimensional input vector $\mathbf{z}$ from a normal distribution, and each row corresponds to one of the dimensions. Moving across a row, we vary the value of dimension z_i from -2 to $+2$ while keeping the other 11 dimensions unchanged. It can be seen that, the redundant dimensions are successfully deactivated. While the only activated dimension (the top left row) controls the unique factor of variation in the dataset.

C. Ablation Study

To demonstrate the effectiveness of our OroJaR, we train a simple GAN (6 layers, the network architectures are shown in Fig. 4.) on the Dsprites dataset under three different settings:

- Firstly, we train the GAN without applying the OroJaR (GAN).
- Secondly, we train the GAN by applying OroJaR to every single intermediate layer (L0 to L4).
- Thirdly, we train the GAN by applying the OroJaR to the first multiple layers (L0∼2 to L0∼4).

Fig. 5 and Table 1 show the qualitative and quantitative comparison among these settings. The results on a single intermediate layer (L0 to L4) show that the earlier layers are more effective in deactivating the redundant dimensions, resulting in better disentanglement. By applying OroJaR to the deeper layer, the benefit of OroJaR to the disentanglement first increases and then sharply decreases. Nonetheless, applying OroJaR to a single intermediate layer is not sufficient to learn a well-disentangled model, and applying OroJaR to the first multiple layers helps learn a better disentangled model. Our OroJaR empirically achieves the best disentanglement performance when D corresponding to the last layer before the last upsampling layer.

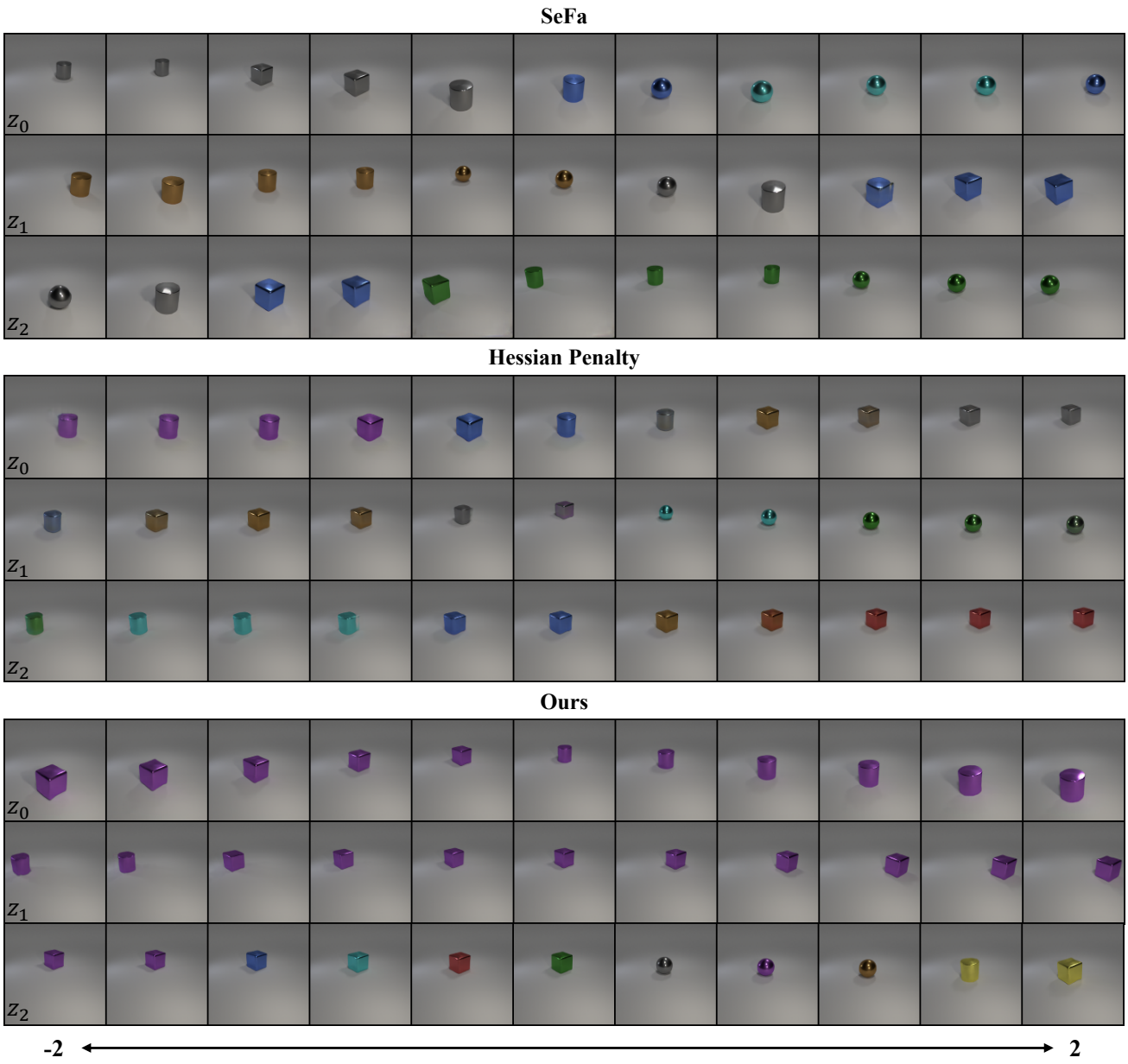

Figure 2: Comparison of disentanglement quality by SeFa [2], Hessian Penalty [1], and our OroJaR on the CLEVR-U dataset. CLEVR-U indicates that we trained the model on CLEVR-Simple (4 factors) by setting the dimension of input to 3. For each method, we randomly sample a 3-dimensional Gaussian vector, and each row corresponds to one of the dimensions. **Top**: In this underparameterized setting, SeFa learns an entangled representation, and each dimension controls all four variations at the same time. **Middle**: Hessian Penalty also entangles the shape with position variation. When moving the object along a direction, the shape of the object is also changed at the same time (the last two rows). **Bottom**: Our method learns a better disentangled result. From top to down, each dimension controls the variations of vertical position, horizontal position, and shape (entangles with color), respectively.

References

[1] William Peebles, John Peebles, Jun-Yan Zhu, Alexei A. Efros, and Antonio Torralba. The hessian penalty: A weak prior for unsupervised disentanglement. In *Proceedings of the European Conference on Computer Vision*, 2020. 1, 3, 4

[2] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 1, 3

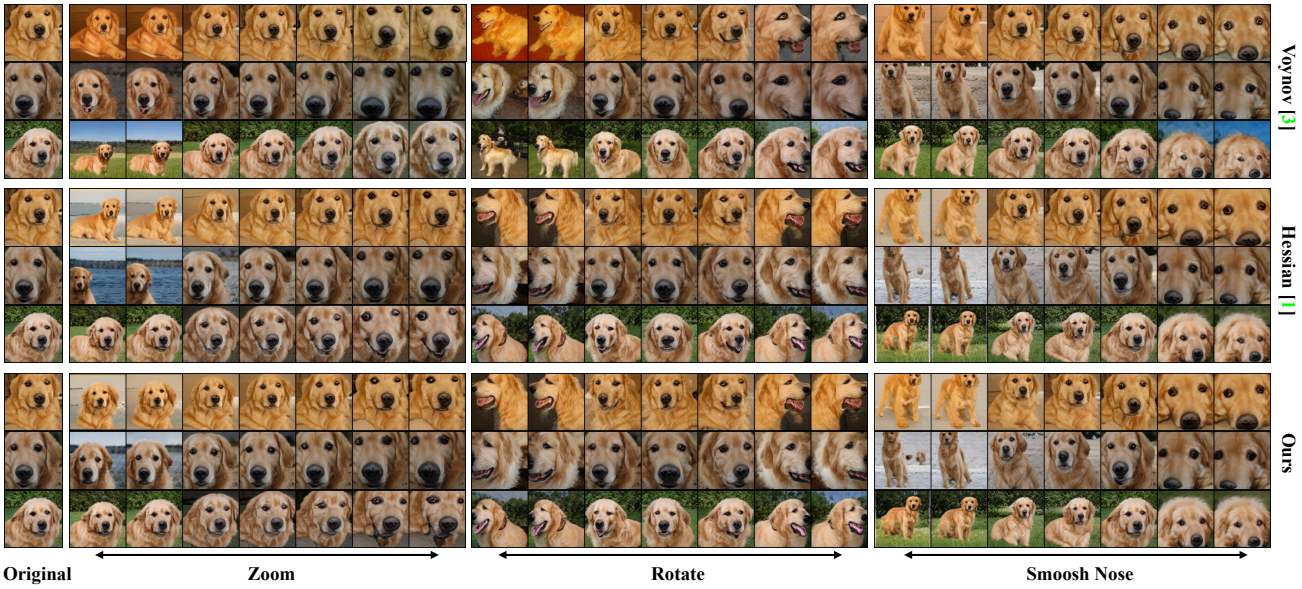

Figure 3: Comparing the quality of latent space editing by our OroJaR, Hessian Penalty [1] and Voynov [3]. Voynov [3] learns entangled rotation factor, and gets a degraded results when rotating the dog to the left side. Hessian Penalty learns a similar rotation and smoosh nose factors with ours, but our method achieves a better zoom editing quality.

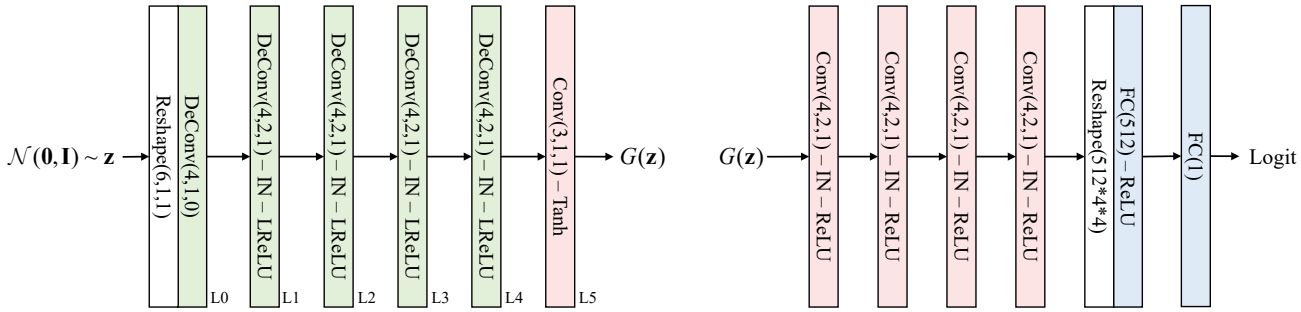

Figure 4: Network architectures of simple GAN used on Dsprites experiments. Conv(k, s, p) and DeConv(k, s, p) denote convolutional layer and transposed convolutional layer where k is kernel size, s is stride and p is padding size. FC(d) denotes fully connected layer with d as output dimension. LReLU denotes the Leaky ReLU nonlinearity.

[3] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *International Conference on Machine Learning*, pages 9786–9796. PMLR, 2020. 1, 4

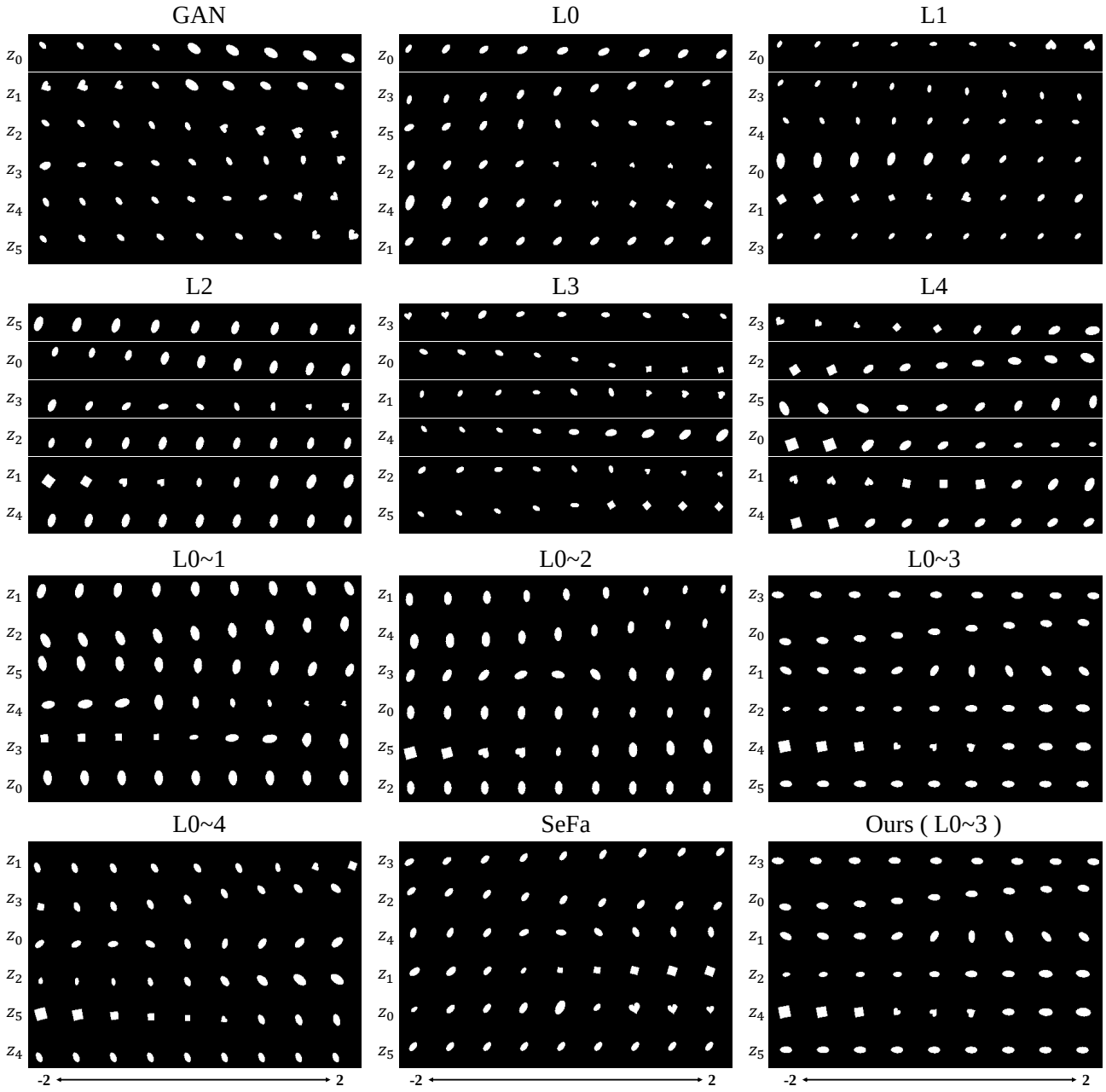

Figure 5: Effectiveness of our OroJaR in disentanglement learning. Lx means OroJaR is applied to the x-th layer, and L0 $\sim$ x means OroJaR is applied to first x layers. We found that earlier layers are more effective in deactivating the redundant dimensions, resulting in better disentanglement. Applying OroJaR to a single intermediate layer is not sufficient to learn a well-disentangled model, and applying OroJaR to first multiple layers helps learn a better disentangled model.