# Supplementary to the ARD-VAE: A Statistical Formulation to Find the Relevant Latent Dimensions of Variational Autoencoders

Surojit Saha
The University of Utah, USA
surojit.saha@utah.edu

Sarang Joshi
The University of Utah, USA
sarang.joshi@utah.edu

Ross Whitaker
The University of Utah, USA
whitaker@cs.utah.edu

## 1. Training of the ARD-VAE

The Algorithm 1 outlines the training of the ARD-VAE. The proposed method uses data in the latent space, $D_{\mathbf{z}}$, to update the parameters of the prior distribution, $p(\mathbf{z} \mid D_{\mathbf{z}})$ (that is not fixed, unlike the regular VAE [7, 12]). The data $D_{\mathbf{z}}$ in the latent space is produced using the Algorithm 2 that takes in as input a subset of the training data, $\mathcal{X}_{\boldsymbol{\alpha}}$.

**Algorithm 1** : **ARD-VAE training**

**Input:** Training samples $\mathcal{X}$, Number of epochs the $D_{\mathbf{z}}$ should lag $u_{D_{\mathbf{z}}}$, Scaling factor $\beta$
**Output:** Encoder and decoder parameters, $\phi$ and $\theta$
Split $\mathcal{X}$ into training, $\mathcal{X}_{train}$, and validation data, $\mathcal{X}_{val}$
Choose a subset $\mathcal{X}_{\boldsymbol{\alpha}}$ at random from $\mathcal{X}_{train}$
Initialize SGD samples $\mathcal{X}_{sgd} = \mathcal{X}_{train} - \mathcal{X}_{\boldsymbol{\alpha}}$
Initialize $\phi$ and $\theta$
Initialize epoch index $e \leftarrow 0$
**for** number of epochs **do**
  **if** $e \mod u_{D_{\mathbf{z}}}$ **then**
    Choose a new subset of the training data $\mathcal{X}_{\boldsymbol{\alpha}}$
    Update SGD samples, $\mathcal{X}_{sgd} = \mathcal{X}_{train} - \mathcal{X}_{\boldsymbol{\alpha}}$
    Update $D_{\mathbf{z}}$ with the samples produced by the Algorithm 2 using the update $\mathcal{X}_{\boldsymbol{\alpha}}$
    Update $\boldsymbol{b}_L$, using the latest $D_{\mathbf{z}}$
  **end if**
  **for** number of minibatch updates **do**
    Sample a minibatch from $\mathcal{X}_{sgd}$
    Update $\phi$ and $\theta$ by optimizing the ARD-VAE objective function
  **end for**
  $e \leftarrow e + 1$
  Shuffle $\mathcal{X}_{sgd}$
**end for**

**Ablation Study on the Size of $\mathcal{X}_{\boldsymbol{\alpha}}$ :** The size of the $\mathcal{X}_{\boldsymbol{\alpha}}$ in Algorithm 1 can be treated as a hyperparameter of the proposed method. Thus, we conduct an ablation study on the size of the $\mathcal{X}_{\boldsymbol{\alpha}}$. In this analysis, we study the effect of the number of samples in $\mathcal{X}_{\boldsymbol{\alpha}}$ on the performance of

**Algorithm 2** Produce $D_{\mathbf{z}}$ using the training subset $\mathcal{X}_{\boldsymbol{\alpha}}$

**Input:** $\mathcal{X}_{\boldsymbol{\alpha}}$
**Output:** $D_{\mathbf{z}}$
Initialize $D_{\mathbf{z}} \leftarrow \phi$.
**for** $\mathbf{x}^{'} \in \mathcal{X}_{\boldsymbol{\alpha}}$ **do**
  $\boldsymbol{\mu}_{\mathbf{x}'}, \boldsymbol{\sigma}^2_{\mathbf{x}'} \leftarrow \mathbf{E}_{\phi}(\mathbf{x}^{'})$
  $\mathbf{z}^{'} \leftarrow \boldsymbol{\mu}_{\mathbf{x}'} + \boldsymbol{\epsilon} \odot \boldsymbol{\sigma}_{\mathbf{x}'}$
  $D_{\mathbf{z}} \leftarrow D_{\mathbf{z}} \cup \mathbf{z}^{'}$
**end for**

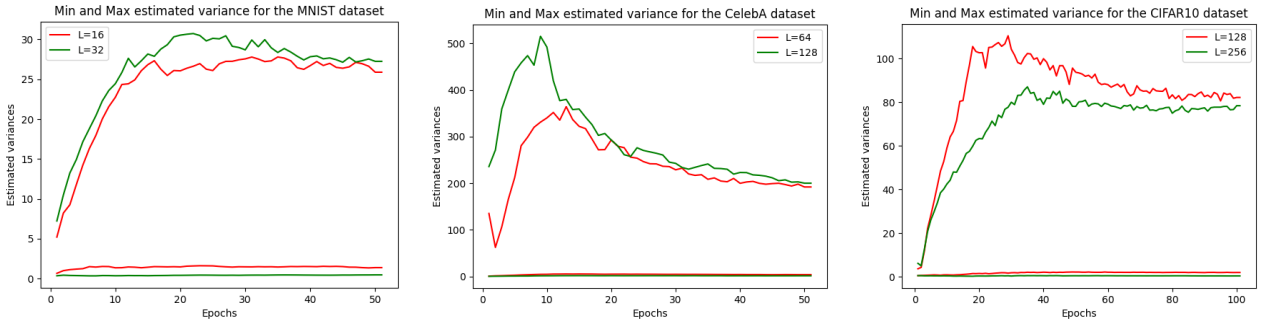
the ARD-VAE in terms of the number of relevant/active dimensions identified and FID score of the generated samples. Different settings of $\mid \mathcal{X}_{\boldsymbol{\alpha}} \mid$ considered in this study are $\mid \mathcal{X}_{\boldsymbol{\alpha}} \mid = \{2K, 5K, 10K, 20K\}$. The ARD-VAE is trained on the MNIST and CIFAR10 datasets with $L = 32$ and $L = 256$, respectively. We have used the neural network architecture in Tab. 4 and followed the optimization strategies discussed in the section 2. We use the hyperparameter $\beta$ as reported in Tab. 6. Compared to other results reported in the results section, the network parameters are initialized using a single seed, as we did not observe much variation in the performance of the ARD-VAE with different initializations.

From the results reported in Tab. 1, we observe negligible variation in the number of relevant dimensions estimated by the ARD-VAE with the size of $\mathcal{X}_{\boldsymbol{\alpha}}$ and there is almost no variation in the FID scores of the generated samples for both datasets. Thus, we conclude the performance of the ARD-VAE is not affected by the number of samples in $\mathcal{X}_{\boldsymbol{\alpha}}$. The ARD-VAE is trained in an unsupervised framework having access to a large amount of data, and it is good to have a representative set that captures the variations in the dataset. Thus, we choose $\mid \mathcal{X}_{\boldsymbol{\alpha}} \mid = 10K$ as a general setting for any random dataset that uses the ARD-VAE.

Considering the size of the training set $(50K)$ of the MNIST and CIFAR10 datasets, $\mid \mathcal{X}_{\boldsymbol{\alpha}} \mid = 10K$ is a large number. This motivated us to evaluate the performance of the ARD-VAE with different settings of $\mid \mathcal{X}_{\boldsymbol{\alpha}} \mid$ on the *large* ImageNet dataset containing $\sim 1.28$ million training sam-

| | $\mid \mathcal{X}_{\boldsymbol{\alpha}} \mid = 2K$ | | $\mid \mathcal{X}_{\boldsymbol{\alpha}} \mid = 5K$ | | $\mid \mathcal{X}_{\boldsymbol{\alpha}} \mid = 10K$ | | $\mid \mathcal{X}_{\boldsymbol{\alpha}} \mid = 20K$ | |
|---|---|---|---|---|---|---|---|---|
| | ACTIVE | FID ↓ | ACTIVE | FID ↓ | ACTIVE | FID ↓ | ACTIVE | FID ↓ |
| MNIST ($L = 32$) | 13 | 21.83 | 12 | 22.04 | 13 | 22.13 | 13 | 24.12 |
| CIFAR10 ($L = 256$) | 116 | 85.15 | 114 | 85.65 | 112 | 85.99 | 112 | 86.83 |

Table 1. The number of ACTIVE dimensions and the FID scores of the generated samples for different sizes of the $\mathcal{X}_{\boldsymbol{\alpha}}$ in the ARD-VAE.




(a) MNIST (b) CelebA (c) CIFAR10

Figure 1. The minimum and maximum variances estimated by the ARD-VAE while training on the MNIST, CelebA and CIFAR10 datasets for multiple latent dimensions. The maximum estimated variances are orders of magnitude higher than the minimum estimated variances.

| $\mid \mathcal{X}_{\boldsymbol{\alpha}} \mid$ | ACTIVE | FID↓ | PRECISION↑ | RECALL↑ | MSE↓ |
|---|---|---|---|---|---|
| $10K$ | 152 | 120.53 | 0.56 | 0.48 | 0.005 |
| $20K$ | 153 | 119.66 | 0.53 | 0.57 | 0.005 |

Table 2. Effect of the size of the $\mid \mathcal{X}_{\boldsymbol{\alpha}} \mid$ on the performance of the ARD-VAE under different metrics for the **large ImageNet dataset**. We observe the ARD-AVE is resilient to the size of the $\mid \mathcal{X}_{\boldsymbol{\alpha}} \mid$ for a sufficiently large size, such as $\mid \mathcal{X}_{\boldsymbol{\alpha}} \mid = 10K$. This observation is consistent with the results on the MNSIT and CIFAR10 datasets in Tab. 1. Therefore, we demonstrate that the configuration of the ARD-VAE used in relatively smaller datasets, such as the MNIST and CIFAR10, seamlessly scales to the large ImageNet dataset.

| Dataset | $L$ $(secs)$ | $2L$ $(secs)$ | $4L$ $(secs)$ |
|---|---|---|---|
| MNIST ($L = 16$) | $0.36 \pm 0.02$ | $0.36 \pm 0.01$ | $0.36 \pm 0.01$ |
| CIFAR10 ($L = 128$) | $0.64 \pm 0.02$ | $0.64 \pm 0.02$ | $0.65 \pm 0.02$ |

Table 3. Time taken (in secs) in the computation of $\boldsymbol{b}_L$ on a 12GB NVIDIA TITAN V to get the updated $\hat{\boldsymbol{\sigma}}^2$ using $\mid \mathcal{X}_{\boldsymbol{\alpha}} \mid = 10K$ (refer to the Algorithm 1). The inference time is indifferent to the size of the latent space, $L$. However, it increases with the complexity of the neural network, such as for the CIFAR10 dataset.

ples. We train the ARD-VAE on the ImageNet dataset with the configuration of $\mid \mathcal{X}_{\boldsymbol{\alpha}} \mid = \{10K, 20K\}$ and evaluate its performance under several metrics, such as the relevant dimensions (ACTIVE), FID score of the generated samples, precision-recall scores, and reconstruction loss (MSE). We have used the neural network architecture in Tab. 4 and followed the optimization strategies discussed in the section 2. From the results reported in Tab. 2, we observe that the performance ARD seamlessly scales to the large ImageNet dataset, and its performance is not affected with more samples in $\mid \mathcal{X}_{\boldsymbol{\alpha}} \mid$. Therefore, we empirically demonstrate the robustness of the setting, $\mid \mathcal{X}_{\boldsymbol{\alpha}} \mid = 10K$, across multiple datasets. We use $\mid \mathcal{X}_{\boldsymbol{\alpha}} \mid = 10K$ for other large datasets (relative to the MNIST and CIFAR10 datasets) studied in this work, e.g., the CelebA ($\sim 200K$), DSprites ($\sim 700K$) and 3D Shapes ($\sim 500K$) datasets.

The parameters of the prior distribution, $p(\mathbf{z} \mid D_{\mathbf{z}})$, are updated every epoch (indicated by $u_{D_{\mathbf{z}}}$ in the Algorithm 1) using samples in the latent space. In Tab. 3, we report the time taken (in secs) on a 12GB NVIDIA TITAN V to estimate the parameters of the prior distribution using $\mid \mathcal{X}_{\boldsymbol{\alpha}} \mid = 10K$ for the MNIST and CIFAR10 datasets. We observe the inference time is indifferent to the size of the latent space for both the datasets. The inference time increases with the complexity of the encoder-decoder architecture used for different datasets, such as the neural network used for the CIFAR10 dataset with more model parameters than the MNIST takes more time to estimate the parameters of the posterior distribution. However, as the parameters are updated *only once* in an epoch, the inference time taken is negligible relative to the training duration. The time taken to estimate the distribution parameters for the ImageNet dataset is similar to the CIFAR10, as we use the same neural network architecture. Thus, the results in Tab. 3, demonstrates the feasibility of training the ARD-VAE on large datasets and bigger latent spaces. Moreover, we could successfully train the ARD-VAE on the ImageNet dataset on a single 12GB NVIDIA TITAN V using the architecture mentioned in Tab. 4 and parameters in Tab. 6.

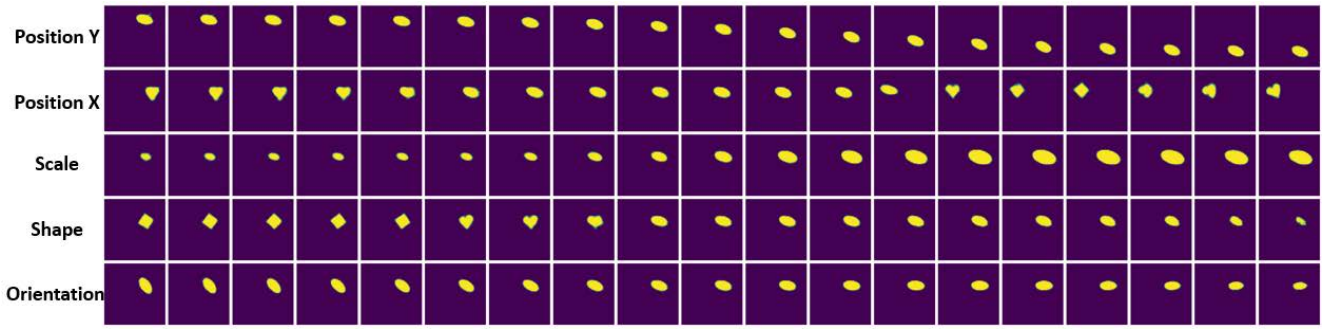



Figure 2. Latent traversal of the DSprites data set [10] in the range $[-3\sigma, 3\sigma]$ using the relevant axes discovered by the ARD-VAE. The latent factors are mentioned in the left column. All latent factors are represented by independent latent axes with slight entanglement of **Shape** and **Position Y**. The MIG score for this model is $\mathbf{0.35}$

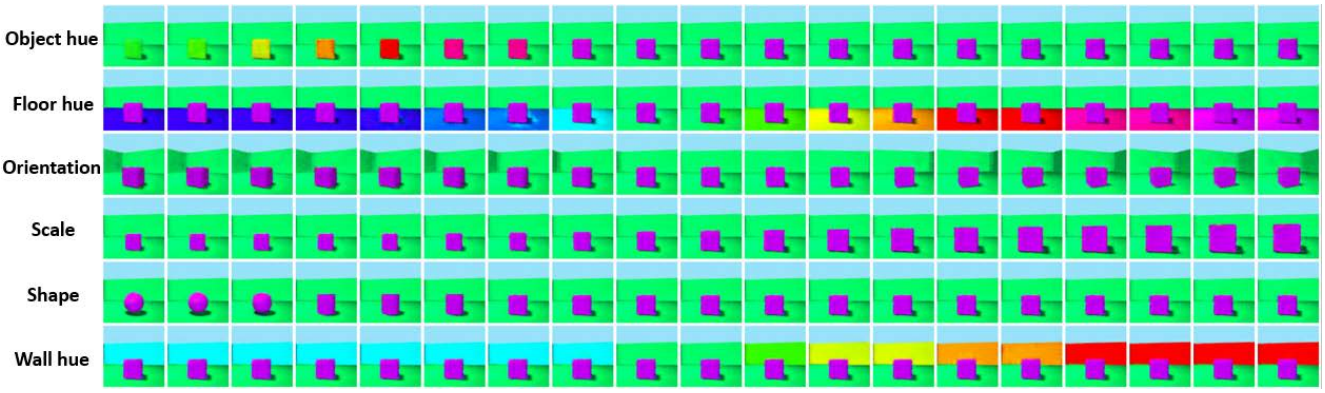



Figure 3. Latent traversal of the 3D Shapes data set [1] in the range $[-3\sigma, 3\sigma]$ using the relevant axes discovered by the ARD-VAE. The latent factors are mentioned in the left column. All latent factors are represented by independent latent axes with almost no overlap between them. The MIG score for this model is $\mathbf{0.84}$

The prior distribution of the ARD-VAE is not fixed, unlike the regular VAE [7, 12]. Thus, we track the minimum and maximum estimated variances for the MNIST, CelebA and CIFAR10 datasets across the training epochs in Fig. 1. The minimum estimated variance is significantly less than the maximum variance across different experimental setup and the maximum estimated variance stabilizes after certain number of epochs, depending on the dataset. For all the datasets, the estimated variances (both the minimum and maximum) are similar for different sizes of the latent space. These figures illustrate the stability in the training of the ARD-VAE across multiple datasets.

The ARD-VAE outperforms the competing methods under different disentanglement metrics on multiple datasets, as reported in Tab. 1 in the main paper. The information encoded by the relevant latent dimensions identified by the ARD-VAE when trained on the DSprites and 3D Shapes are shown in Fig. 2 and Fig. 3, respectively. In both images, Fig. 2 and Fig. 3, we traverse each relevant latent axis in the range $[-3\sigma, 3\sigma]$ to interpret the variability explained by the axes. From these images, we conclude that the ARD-VAE identifies all factors of variation present in both the DSprites and 3D Shapes datasets. In Fig. 4, we traverse all the latent axes, $L = 10$, used in the training of the ARD-AVE on the 3D Shapes dataset, sorted by the relevance score proposed in the paper. We observe that the decoder produces no variability in output in response to the deviations along *a few latent axes* with lower relevance score, highlighted within the red bounding box. This visualization demonstrates the presence of *irrelevant* or *superfluous* latent dimensions that the ARD-VAE correctly identifies.

## 2. Experimental Settings

In the neural network architectures reported in Tab. 4, Tab. 5, CONV$n$ and TRANSCONV$n$ define convolution and

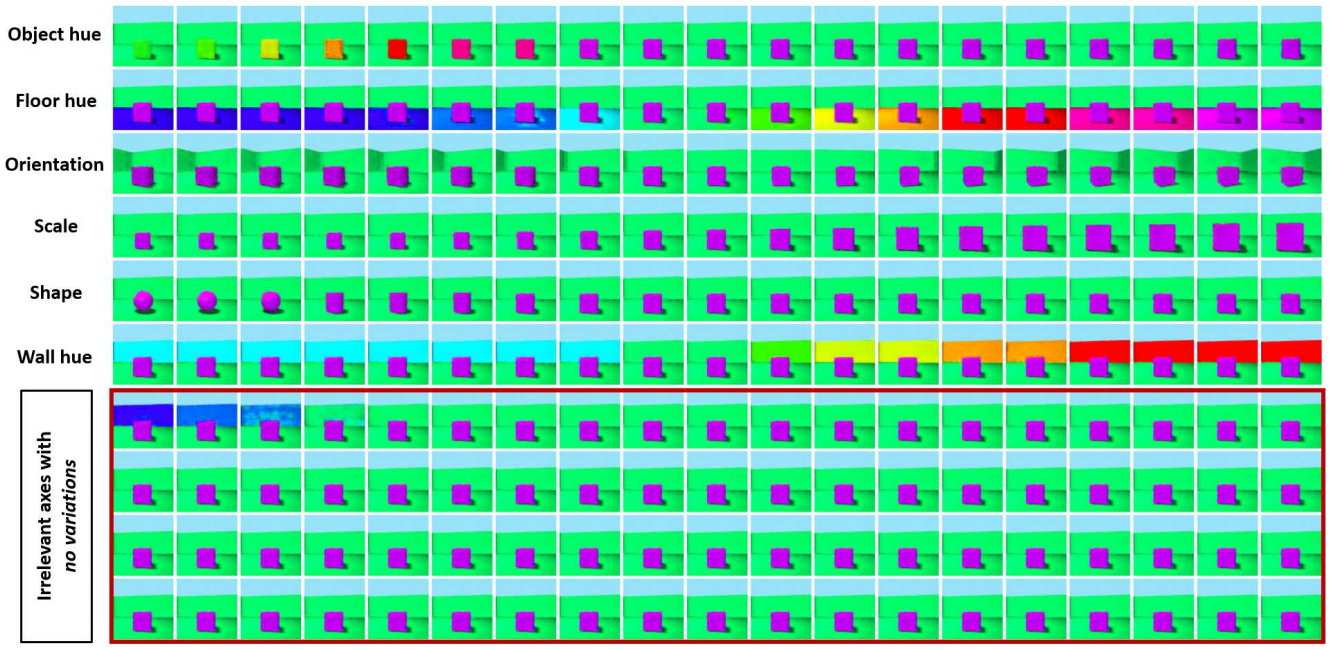



Figure 4. Latent traversal of the 3D Shapes data set [1] in the range $[-3\sigma, 3\sigma]$ on all the latent axes, $L = 10$, used in the training of the ARD-AVE on the 3D Shapes dataset, sorted by the relevance score proposed in the paper. The latent factors are mentioned in the left column for the relevant axes (total 6), and additional axes (total 4) are highlighted within the red bounding box that shows no variability in output in response to deviations along these axes. This elucidates our hypothesis about the behavior of *irrelevant* latent axes.

| | MNIST | CelebA | CIFAR10 & ImageNet |
|---|---|---|---|
| Encoder: | $\mathbf{x} \in \mathbb{R}^{32\times32\times1}$ | $\mathbf{x} \in \mathbb{R}^{64\times64\times3}$ | $\mathbf{x} \in \mathbb{R}^{32\times32\times3}$ |
| | CONV64 $\rightarrow$ BN $\rightarrow$ RELU | CONV64 $\rightarrow$ BN $\rightarrow$ RELU | CONV128 $\rightarrow$ BN $\rightarrow$ RELU |
| | CONV128 $\rightarrow$ BN $\rightarrow$ RELU | CONV128 $\rightarrow$ BN $\rightarrow$ RELU | CONV256 $\rightarrow$ BN $\rightarrow$ RELU |
| | CONV256 $\rightarrow$ BN $\rightarrow$ RELU | CONV256 $\rightarrow$ BN $\rightarrow$ RELU | CONV512 $\rightarrow$ BN $\rightarrow$ RELU |
| | CONV512 $\rightarrow$ BN $\rightarrow$ RELU | CONV512 $\rightarrow$ BN $\rightarrow$ RELU | CONV1024 $\rightarrow$ BN $\rightarrow$ RELU |
| | FLATTEN$_{2\times2\times512}$ $\rightarrow$ FC$_{k\times16}$ $\rightarrow$ NONE | FLATTEN$_{4\times4\times512}$ $\rightarrow$ FC$_{k\times64}$ $\rightarrow$ NONE | FLATTEN$_{2\times2\times1024}$ $\rightarrow$ FC$_{k\times128}$ $\rightarrow$ NONE |
| Decoder: | $\mathbf{z} \in \mathbb{R}^{16}$ $\rightarrow$ FC$_{2\times2\times512}$ | $\mathbf{z} \in \mathbb{R}^{64}$ $\rightarrow$ FC$_{8\times8\times512}$ | $\mathbf{z} \in \mathbb{R}^{128}$ $\rightarrow$ FC$_{8\times8\times1024}$ |
| | TRANSCONV256 $\rightarrow$ BN $\rightarrow$ RELU | TRANSCONV256 $\rightarrow$ BN $\rightarrow$ RELU | TRANSCONV512 $\rightarrow$ BN $\rightarrow$ RELU |
| | TRANSCONV128 $\rightarrow$ BN $\rightarrow$ RELU | TRANSCONV128 $\rightarrow$ BN $\rightarrow$ RELU | TRANSCONV256 $\rightarrow$ BN $\rightarrow$ RELU |
| | TRANSCONV64 $\rightarrow$ BN $\rightarrow$ RELU | TRANSCONV64 $\rightarrow$ BN $\rightarrow$ RELU | TRANSCONV3 $\rightarrow$ SIGMOID |
| | TRANSCONV1 $\rightarrow$ SIGMOID | TRANSCONV3 $\rightarrow$ TANH | |

Table 4. Encoder and decoder architectures used by all the methods for the MNIST, CelebA and CIFAR10 datasets [4, 14].

transpose convolution operation, respectively, with $n$ filters in the output. We have used $4 \times 4$ filters for all the datasets. The transpose convolution filters use a stride size of 2 except for the last layer of the decoders used in the CelebA, CIFAR10 and ImageNet datasets. We represent the fully connected layers as FC$_{k\times n}$ with $k \times n$ nodes, where $k = 1$ for all the methods, except the VAE and $\beta$-TCVAE that use $k = 2$. Activation functions used in the networks are ReLU (RELU), Leaky ReLU (LRELU), sigmoid (SIGMOID), and hyperbolic tangent (TANH). Input is in the range $[0, 1]$ for all the datasets except CelebA, for which the input is mapped to the range $[-1, 1]$. The encoder-decoder architectures of the MaskAAE [11] and GECO-$L_0$-ARM-VAE [2] are obtained from the respective papers.

We use the Adam optimizer in all experiments (learning rate set to $5e-04$) with a learning rate scheduler (ReduceLROnPlateau) that reduces the learning rate by $0.5$ if the validation loss does not improve for a maximum of $10$ epochs except the MaskAAE and GECO-$L_0$-ARM-VAE due to their sensitivity in optimization of the trainable parameters. Moreover, the MaskAAE uses a specific training recipe. All the methods are trained for $50$, $50$, $100$, and $50$ epochs for the MNIST, CelebA, CIFAR10, and ImageNet datasets, respectively, with a few exceptions for the MNIST dataset. The VAE, $\beta$-TCVAE, GECO-$L_0$-ARM-VAE, and MaskAEE are trained for 100 epochs for the MNIST dataset

| | DSprites | 3D Shapes |
|---|---|---|
| Encoder: | $\mathbf{x} \in \mathbb{R}^{64\times64\times1}$ | $\mathbf{x} \in \mathbb{R}^{64\times64\times3}$ |
| | CONV32 $\rightarrow$ RELU | CONV32 $\rightarrow$ RELU |
| | CONV32 $\rightarrow$ RELU | CONV32 $\rightarrow$ RELU |
| | CONV64 $\rightarrow$ RELU | CONV64 $\rightarrow$ RELU |
| | CONV64 $\rightarrow$ RELU | CONV64 $\rightarrow$ RELU |
| | FLATTEN$_{4\times4\times64}$ $\rightarrow$ FC$_{k\times6}$ $\rightarrow$ NONE | FLATTEN$_{4\times4\times64}$ $\rightarrow$ FC$_{k\times6}$ $\rightarrow$ NONE |
| Decoder: | $\mathbf{z} \in \mathbb{R}^6 \rightarrow$ FC$_{4\times4\times64}$ | $\mathbf{z} \in \mathbb{R}^6 \rightarrow$ FC$_{4\times4\times64}$ |
| | TRANSCONV64 $\rightarrow$ RELU | TRANSCONV64 $\rightarrow$ RELU |
| | TRANSCONV32 $\rightarrow$ RELU | TRANSCONV32 $\rightarrow$ RELU |
| | TRANSCONV32 $\rightarrow$ RELU | TRANSCONV32 $\rightarrow$ RELU |
| | TRANSCONV1 $\rightarrow$ NONE | TRANSCONV3 $\rightarrow$ SIGMOID |

Table 5. Encoder and decoder architectures used by all the methods for the DSprites and 3D Shapes datasets [6, 9].

| Method | Parameters | MNIST | CelebA | CIFAR10 | ImageNet | DSprites | 3D Shapes |
|---|---|---|---|---|---|---|---|
| $\beta$-TCVAE | $\beta$: | 2 | 2 | 2 | 2 | 5 | 5 |
| DIP-VAE-I | $(\lambda_{od}, \lambda_d)$: | NA | NA | NA | NA | $(10, 100)$ | $(10, 100)$ |
| DIP-VAE-II | $(\lambda_{od}, \lambda_d)$: | NA | NA | NA | NA | $(10, 10)$ | $(10, 10)$ |
| WAE | RECONS-SCALAR: | 0.05 | 0.05 | 0.05 | 0.05 | NA | NA |
| WAE | $\beta$: | 10 | 100 | 100 | 100 | NA | NA |
| RAE | $\beta$: | $1e-04$ | $1e-04$ | $1e-03$ | $1e-03$ | $1e-04$ | $1e-04$ |
| RAE | DEC-L2-REG: | $1e-07$ | $1e-07$ | $1e-06$ | $1e-06$ | $1e-06$ | $1e-06$ |
| GECO-$L_0$-ARM-VAE | $\tau$: | 10.0 | 300.0 | 25.0 | 15.0 | NA | NA |
| ARD-VAE | $\beta$: | 0.5 | 1.0 | 0.05 | 0.05 | 5.0 | 5.0 |

Table 6. Optimization settings for different methods.

| DSprites | | | 3D Shapes | | |
|---|---|---|---|---|---|
| INITIAL | ACTIVE | GT | INITIAL | ACTIVE | GT |
| 10 | $5.80 \pm 0.40$ | 6 | 10 | $6.40 \pm 0.49$ | 6 |
| 15 | $6.00 \pm 0.00$ | 6 | 15 | $6.40 \pm 0.49$ | 6 |
| 20 | $5.80 \pm 0.40$ | 6 | 20 | $6.40 \pm 0.49$ | 6 |
| 30 | $5.80 \pm 0.40$ | 6 | 30 | $6.20 \pm 0.40$ | 6 |

Table 7. The number of the active dimensions (ACTIVE) estimated by the ARD-VAE matches closely with the ground truth (GT).

as it improved the model performance. In the disentanglement analysis, all the methods are trained for 35 and 60 epochs [9] for the DSprites and 3D Shapes datasets, respectively.

We use a batch size of 100 for training all the methods, except the MaskAAE, which is trained using a batch size of 64 (to maintain consistency with their implementation). All the hyperparameters of the MaskAAE are set according to the GitHub repo in https://github.com/arnabkmondal/MaskAAE for all the datasets. We have tuned the hyperparameters of the MaskAAE over a series (i.e., dozens) of experiments, and no better results could be obtained. For the GECO-$L_0$-ARM-VAE, we have used the code shared by the authors [2]. Only the target reconstruction loss ($\tau$) in GECO-$L_0$-ARM-VAE was the hyperparameter in our analysis. For the DIP-VAE-I and DIP-VAE-II, we have followed the suggested hyperparameters in the paper [8]. For the $\beta$-TCVAE, we have empirically determined the strength of the regularization loss for different data sets as we could not find them in the literature. We set $\beta = 2$ for the MNIST, CelebA, and CIFAR10 data sets as higher $\beta$ resulted in poor reconstruction. Table 6 reports the specific hyperparameters for different methods. The $u_{D_\mathbf{z}}$ in the algorithm 1 is set to 1 and $\mid \mathcal{X}_{\boldsymbol{\alpha}} \mid = 10K$ for all the datasets studied in this work.

## 3. Results

In this section, we report the results on the CelebA dataset for all the competing methods, the reconstruction loss on different datasets, an ablation study on the effect of the relevance score estimation method on the performance of the ARD-VAE, and demonstrate the use of the relevance score estimation method in the determination of the number of relevant/active dimensions for other competing methods studied in this work.

**Results on the synthetic dataset :** We know the number of latent factors used in the generation of the synthetic datasets, the DSprites [10] and 3D Shapes [1], which is 6 for both datasets. Thus, the number of *known* latent factors in these datasets serves as the *ground truth* for the number of relevant (or ACTIVE) dimensions estimated by the ARD-VAE. In this experiment, we set the initial size of the latent space, $L = 10, 15, 20, 30$, and train the ARD-VAE

| METHOD | DSPRITES | | | 3D SHAPES | | |
|---|---|---|---|---|---|---|
| | FACTORVAE METRIC ↑ | MIG ↑ | ACTIVE | FACTORVAE METRIC ↑ | MIG ↑ | ACTIVE |
| $\beta$-TCVAE ($L = 10$) | **81.15** | 0.23 | 10 | 84.29 | 0.48 | 10 |
| ARD-VAE ($L = 10$) | 66.63 | **0.26** | 6 | **91.40** | **0.84** | 6 |

Table 8. Disentanglement scores of competing methods, where we initialize the model parameters of the different methods with the *same* seed for multiple datasets. In this experiment, we train the $\beta$-TCVAE (a baseline method comparable to the ARD-VAE in Tab. 1 in the main paper) with the latent dimensions $L = 10$ and compare its performance with the ARD-VAE under the same setting. The ACTIVE indicates the number of the latent dimensions used by different methods to compute the metric scores (higher is better). We train the ARD-VAE with $L = 10$ and find the number of relevant dimensions as 6 (ACTIVE). The **best** score is in **bold**. The ARD-VAE mostly outperforms the $\beta$-TCVAE (similar to Tab. 1 in the main paper), even with $L = 10$.

| | CelebA ($L = 64$) | | | | ImageNet ($L = 256$) | | | |
|---|---|---|---|---|---|---|---|---|
| | ACTIVE | FID↓ | PRECISION↑ | RECALL↑ | ACTIVE | FID↓ | PRECISION↑ | RECALL↑ |
| VAE | 64 | $\underline{49.89 \pm 0.57}$ | $0.79 \pm 0.03$ | $\underline{0.75 \pm 0.03}$ | 256 | $180.44 \pm 0.69$ | $0.12 \pm 0.01$ | $0.31 \pm 0.04$ |
| $\beta$-TCVAE | 64 | $50.14 \pm 0.78$ | $0.78 \pm 0.02$ | $0.70 \pm 0.05$ | 256 | $226.27 \pm 0.48$ | $0.04 \pm 0.01$ | $0.25 \pm 0.01$ |
| RAE | 64 | $\mathbf{48.81 \pm 1.02}$ | $\underline{0.81 \pm 0.02}$ | $\mathbf{0.77 \pm 0.04}$ | 256 | $226.70 \pm 30.14$ | $0.08 \pm 0.05$ | $0.11 \pm 0.05$ |
| WAE | 64 | $72.01 \pm 2.26$ | $0.64 \pm 0.05$ | $\underline{0.75 \pm 0.02}$ | 256 | $278.32 \pm 5.83$ | $0.03 \pm 0.00$ | $0.04 \pm 0.08$ |
| GECO-$L_0$-ARM-VAE | $35.60 \pm 3.07$ | $294.97 \pm 28.45$ | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ | $124.00 \pm 9.34$ | $\underline{177.40 \pm 40.21}$ | $\underline{0.27 \pm 0.09}$ | $\underline{0.26 \pm 0.10}$ |
| MaskAAE | $5.4 \pm 0.80$ | $333.40 \pm 10.91$ | $0.01 \pm 0.02$ | $0.00 \pm 0.00$ | $0.0 \pm 0.00$* | — | — | — |
| ARD-VAE | $53.40 \pm 0.49$ | $50.73 \pm 0.29$ | $\mathbf{0.85 \pm 0.02}$ | $0.73 \pm 0.02$ | $152.0 \pm 1.1$ | $\mathbf{121.21 \pm 1.16}$ | $\mathbf{0.54 \pm 0.02}$ | $\mathbf{0.51 \pm 0.03}$ |

* The MaskAAE collapsed all the dimensions for the ImageNet dataset after 30 epochs. Thus, we skip the evaluation of the MaskAAE for the ImageNet dataset.

Table 9. The FID and precision-recall scores of the generated samples, along with the number of latent dimensions (ACTIVE) used by the competing methods to compute different metric scores for the CelebA and ImageNet datasets. The **best** score under a metric is in **bold** and the $\underline{\text{second best}}$ score is $\underline{\text{underlined}}$. The ARD-VAE outperforms other methods *by far* on the ImageNet dataset and is comparable to the competing methods on the CelebA dataset.

to estimate the number of relevant axes for both datasets. From the results reported in Tab. 7, we observe the number of active dimensions estimated by the ARD-VAE closely matches the ground truth for different initial bottleneck dimensions.

In the disentanglement analysis of the synthetic datasets (discussed in the main paper), we leverage the information of the number of the *known latent factors* of the DSprites [10] and Shapes3D [1] dataset to set the size of the latent space ($L = 6$) for the baseline methods. However, we train the ARD-VAE with some additional latent dimensions ($L = 10$), such that it identifies the true generative factors of the datasets and discards the unnecessary axes from the set of relevant axes. For a fair comparison, we train the $\beta$-TCVAE, a baseline method comparable to the ARD-VAE in Tab. 1 in the main paper, with the latent dimensions $L = 10$ and compare its performance with the ARD-VAE under the same setting. The result of this ablation study is reported in Tab. 8. Even under this setting, the performance of the ARD-AVE is better than the $\beta$-TCVAE under all the scenarios but the FactorVAE metric on the DSprites dataset.

**Results on the CelebA and ImageNet dataset :** In this analysis, we report the FID [5] and precision-recall [13] scores of the competing methods in Tab. 9. For the CelebA dataset, the performance of the competing methods are comparable with small variations. The ARD-VAE could achieve the performance of the RAE with $\sim$54 latent dimensions, which is 10 dimensions less. Similar to the MNIST and CIFAR10 dataset, we could not train the GECO-$L_0$-ARM-VAE and MaskAAE on the CelebA dataset using the neural network architecture reported in Tab. 4.

We evaluate the scalability and robustness of the ARD-AVE, and we train the ARD-VAE on the ImageNet dataset [3], i.e., a *complex* and *larger* dataset. In our analysis, we train the ARD-VAE along with the baseline methods on the ImageNet dataset with the image resolution of $32 \times 32$. The comparison of the proposed method with the competing methods under several evaluation metrics is reported in Tab. 9. The ARD-AVE compares favorably to all other methods by a *large margin* and *prunes* more than 100 latent dimensions to model the complex dataset. The MaskAAE collapsed all the latent dimensions on the ImageNet dataset when trained using the neural network architecture reported in Tab. 4. The number of relevant dimensions estimated by the ARD-VAE are consistent even when trained with different sizes of the latent space, such as $L = 512, 1024$ (refer to Tab. 11). We could successfully train the ARD-VAE on the ImageNet dataset on a single 12GB NVIDIA TITAN V using the architecture mentioned in Tab. 4 and optimization parameters in Tab. 6. The training for 50 epochs with $\mid \mathcal{X}_{\boldsymbol{\alpha}} \mid = 10K$ took around $\sim$ 17 hours. Therefore, the training of the ARD-VAE on the ImageNet dataset demonstrates the *stability, robustness*, and *effectiveness* of the proposed method in modeling a large dataset with a lot of variability.

**Reconstruction loss of the competing methods:** In Tab. 10, we report the mean-square error (MSE) loss per

| Method | MNIST ($L = 16$) ↓ | CelebA ($L = 64$) ↓ | CIFAR10 ($L = 128$) ↓ | ImageNet ($L = 256$) ↓ |
|---|---|---|---|---|
| VAE | $0.012 \pm 0.000$ | $\underline{0.021 \pm 0.000}$ | $0.016 \pm 0.000$ | $0.018 \pm 0.000$ |
| $\beta$-TCVAE | $0.018 \pm 0.000$ | $0.024 \pm 0.000$ | $0.021 \pm 0.000$ | $0.022 \pm 0.000$ |
| RAE | $\mathbf{0.003 \pm 0.000}$ | $\mathbf{0.020 \pm 0.000}$ | $\mathbf{0.006 \pm 0.000}$ | $\mathbf{0.003 \pm 0.000}$ |
| WAE | $\underline{0.004 \pm 0.000}$ | $\mathbf{0.020 \pm 0.000}$ | $\underline{0.007 \pm 0.000}$ | $0.007 \pm 0.000$ |
| GECO-$L_0$-ARM-VAE | $\underline{0.004 \pm 0.001}$ | $0.029 \pm 0.002$ | $0.011 \pm 0.001$ | $0.006 \pm 0.000$ |
| MaskAAE | $0.091 \pm 0.091$ | $0.093 \pm 0.093$ | $0.199 \pm 0.199$ | —* |
| ARD-VAE | $0.008 \pm 0.000$ | $\underline{0.021 \pm 0.000}$ | $\mathbf{0.006 \pm 0.000}$ | $\underline{0.005 \pm 0.000}$ |

* The MaskAAE collapsed all the dimensions for the CIFAR10 dataset after 20 epochs. Thus, we skip the evaluation of the MaskAAE for the CIFAR10 dataset.

Table 10. MSE per pixel of the competing methods (averaged over 5 different runs) on the benchmark datasets (lower is better). The **best** score is in **bold**, and the second best score is underlined. The number of ACTIVE dimensions of the GECO-$L_0$-ARM-VAE, MaskAAE and ARD-VAE (refer to Tab. 13) are used for computing the MSE. However, for the remaining methods we use all the initial ($L$) dimensions.

| METHOD | MNIST ($L = 16$) | | | CIFAR10 ($L = 128$) | | | ImageNet ($L = 256$) | | |
|---|---|---|---|---|---|---|---|---|---|
| | ACTIVE | ARD-VAE-ALL | ARD-VAE | ACTIVE | ARD-VAE-ALL | ARD-VAE | ACTIVE | ARD-VAE-ALL | ARD-VAE |
| $L$ | $12.8 \pm 0.40$ | $21.94 \pm 0.67$ | $22.24 \pm 0.57$ | $105.8 \pm 1.33$ | $85.05 \pm 0.84$ | $87.56 \pm 1.21$ | $152.0 \pm 1.1$ | $107.9 \pm 1.08$ | $121.21 \pm 1.16$ |
| $2L$ | $12.6 \pm 0.49$ | $21.97 \pm 0.23$ | $22.30 \pm 0.33$ | $116.4 \pm 3.72$ | $83.73 \pm 2.76$ | $86.50 \pm 1.43$ | $163.0 \pm 1.41$ | $107.02 \pm 1.34$ | $123.34 \pm 1.11$ |
| $4L$ | $12.4 \pm 0.49$ | $22.30 \pm 0.48$ | $22.31 \pm 0.58$ | $117.8 \pm 15.04$ | $84.92 \pm 0.92$ | $87.88 \pm 1.80$ | $161.0 \pm 2.28$ | $107.78 \pm 0.64$ | $123.30 \pm 0.51$ |

Table 11. In this experiment, we assess the impact of the *pruning* latent dimensions in the ARD-VAE by comparing the FID scores over multiple datasets. We compare the FID scores (of the generated samples) of the proposed ARD-VAE with its variant using all the latent axes, referred to herein as ARD-VAE-ALL, to produce the metric scores. The number of relevant latent dimensions the ARD-VAE uses are reported as ACTIVE. The ARD-VAE produces consistent estimates of the ACTIVE dimensions and FID scores for different values of $L$.

pixel on the MNIST, CelebA, CIFAR10, and ImageNet datasets. The number of ACTIVE dimensions of the GECO-$L_0$-ARM-VAE, MaskAAE and ARD-VAE (refer to Tab. 13 and Tab. 9) are used for computing the MSE. However, for the remaining methods we use all the latent ($L$) dimensions. The MaskAAE collapsed all the latent dimensions on the ImageNet dataset, and,thus, we could not compute the MSE. Though the GECO-$L_0$-ARM-VAE has a comparable MSE loss, it failed to model the data distribution. taset. The choice of $\beta$ for the ARD-VAE produces comparable MSE, except the MNIST dataset. This is due to the lack of extensive hyperparameter tuning in the ARD-VAE. However, that did not impact the performance of the ARD-VAE significantly.

**Ablation study on the relevance score estimation:** In this work, we proposed a method to compute the relevancy of latent axes and used the same to estimate the number of active dimensions that are sufficient to model a data distribution. However, it is important to evaluate the performance of the ARD-VAE using all the latent axes. To this end, we compute the FID scores of the ARD-VAE on the MNIST, CIFAR10, and the ImageNet datasets trained on latent spaces of different sizes for an individual dataset, shown in Tab. 11. The ARD-VAE that uses all the latent axes is dubbed as the ARD-VAE-ALL, i.e., it does not *prunes* unnecessary/superfluous dimensions. The ACTIVE in Tab. 11 indicates the number of relevant dimensions used by the ARD-VAE for the evaluation of the FID scores under different settings. This analysis serves a *proxy* to the loss of information due to the *pruning* of *irrelevant* dimensions.

We observe the FID scores computed using the relevant axes (ACTIVE) are marginally higher than the scores computed using all the axes (ALL). However, we get rid of many redundant latent axes using the relevance score estimation technique. For e.g., the FID score for the CIFAR10 (slightly higher) with an initial dimension of $4L = 512$ is produced using *only* $\sim$118 latent axes. Similarly, for the ImageNet dataset, the ARD-VAE prunes $\sim 863$ irrelevant dimensions (estimated using the proposed relevance score) when trained with an initial dimension of $4L = 1024$ *without a significant change* in the FID score. This experiment illustrates that the modeling of the data distributions by the ARD-VAE using *only* the relevant latent axes *preserves most of the information* compared to the ARD-VAE-ALL using all latent axes, as indicated by a small degradation in the FID scores.

To better understand the loss of information due to the pruning of latent dimensions in the ARD-VAE, we compare the ARD-VAE with the *best* performing baseline method determined in terms of the FID score. The best baseline method considers all the latent dimensions used in training the model. Comparing the FID scores reported in Tab. 12, we observe slightly higher FID scores for the ARD-VAE on the MNIST and CelebA dataset, indicating some loss of information relative to the baseline. However, the ARD-VAE outperforms the *best* baseline on the challenging CIFAR10

| METHOD | MNIST ($L = 16$) | | CelebA ($L = 64$) | | CIFAR10 ($L = 128$) | | ImageNet ($L = 256$) | |
|---|---|---|---|---|---|---|---|---|
| | ACTIVE | FID↓ | ACTIVE | FID↓ | ACTIVE | FID↓ | ACTIVE | FID ↓ |
| ***Best*** method among the baseline methods | 16 | $\mathbf{18.79 \pm 0.31}$ | 64 | $\mathbf{48.81 \pm 1.02}$ | 128 | $94.34 \pm 1.58$ | $124.00 \pm 9.34$ | $177.40 \pm 40.21$ |
| ARD-VAE | $12.80 \pm 0.40$ | $22.24 \pm 0.57$ | $53.40 \pm 0.49$ | $50.73 \pm 0.29$ | $105.80 \pm 1.33$ | $\mathbf{87.56 \pm 1.21}$ | $152.0 \pm 1.10$ | $\mathbf{121.21 \pm 1.16}$ |

Table 12. In this analysis, we assess the information loss due to the pruning of latent dimensions in the ARD-VAE by comparing its FID scores (of the generated samples) with the best method among the baseline methods (studied in this work) over multiple datasets. The number of latent dimensions used by competing methods are reported as ACTIVE. The **best** FID score is in **bold**.

| | MNIST ($L = 16$) | | CelebA ($L = 64$) | | CIFAR10 ($L = 128$) | |
|---|---|---|---|---|---|---|
| | ACTIVE | FID↓ | ACTIVE | FID↓ | ACTIVE | FID ↓ |
| VAE | $10.20 \pm 0.40$ | $29.33 \pm 0.41$ | $55.20 \pm 0.40$ | $50.79 \pm 0.64$ | $26.40 \pm 0.49$ | $155.13 \pm 1.45$ |
| $\beta$-TCVAE | $7.40 \pm 0.49$ | $51.35 \pm 0.95$ | $51.40 \pm 0.49$ | $51.61 \pm 0.76$ | $16.20 \pm 0.40$ | $186.03 \pm 1.82$ |
| RAE | $16.00 \pm 0.00$ | $\mathbf{18.85 \pm 0.43}$ | $63.00 \pm 0.00$ | $\mathbf{49.06 \pm 1.16}$ | $125.00 \pm 0.00$ | $\underline{92.66 \pm 1.61}$ |
| WAE | $16.00 \pm 0.00$ | $24.78 \pm 0.77$ | $63.00 \pm 0.00$ | $61.18 \pm 1.60$ | $127.00 \pm 0.00$ | $136.79 \pm 1.35$ |
| GECO-$L_0$-ARM-VAE | $10.00 \pm 1.10$ | $304.75 \pm 64.29$ | $35.60 \pm 3.07$ | $294.97 \pm 28.45$ | $68.00 \pm 2.97$ | $320.75 \pm 45.09$ |
| MaskAAE | $9.80 \pm 1.60$ | $144.92 \pm 16.80$ | $5.4 \pm 0.80$ | $333.40 \pm 10.91$ | $3.80 \pm 0.40$ | $298.30 \pm 8.44$ |
| ARD-VAE | $12.80 \pm 0.40$ | $\underline{22.24 \pm 0.57}$ | $53.40 \pm 0.49$ | $\underline{50.73 \pm 0.29}$ | $105.80 \pm 1.33$ | $\mathbf{87.56 \pm 1.21}$ |

Table 13. The FID scores of the generated samples along with the number of (ACTIVE) latent dimensions used by the competing methods. The **best** FID score is in **bold** and the second best is underlined. The ACTIVE dimensions change for the VAE, $\beta$-TCVAE, RAE and WAE.

and ImageNet datasets. Therefore, using extensive comparisons, we demonstrate the robustness and generalization capability of the ARD-VAE compared to the regular VAE and its variants.

**Relevant axes for the VAE using the Jacobian:** In this experiment we estimate the importance of the latent axes of a trained VAE using 1.

$$\mathbf{w}_{\hat{\boldsymbol{\sigma}}} = \sum_{k=1}^{D} \frac{1}{N} \sum_{i=1}^{N} \|\mathbb{J}_i\|^2, \tag{1}$$

where $\mathbb{J} = \left[ \frac{\partial \hat{\mathbf{x}}}{\partial \mu_1} \cdots \frac{\partial \hat{\mathbf{x}}}{\partial \mu_L} \right] \in \mathbb{R}^{D \times L}$ is the Jacobian matrix.

To get a reliable estimate of the weight vector $\mathbf{w}_{\hat{\boldsymbol{\sigma}}}$ using 1, we compute the Jacobian for multiple data samples, which is typically the size of a minibatch (100) in this work.

This estimate is used to determine the relevant axes for the VAE, $\beta$-TCVAE, RAE, and WAE and the selected axes are used for model evaluations as shown in Table 13. The relevant dimensions of the VAE and $\beta$-TCVAE are less than the initial dimension $L$ for the MNIST and CelebA dataset with comparable estimates of the FID scores. However, similar to the GECO-$L_0$-ARM-VAE and MaskAAE, majority of the dimensions collapses for the complex CIFAR dataset. This experiment demonstrates the robustness of the proposed ARD-VAE across different evaluation scenarios. There is no change in the number of active latent dimensions from $L$ for the RAE and WAE as both methods match aggregate posterior distributions, unlike VAEs.

**Size of the latent space for modeling a new dataset:** We know, the size of the latent space of the VAE and its variants for a dataset and a given autoencoder architecture

| Method | $L$ | $2L$ | $4L$ |
|---|---|---|---|
| RAE ($L = 128$) | 104.90 | 116.94 | 121.50 |
| ARD-VAE ($L = 128$) | 85.25 | 84.55 | 84.60 |

Table 14. The FID scores of the RAE (second best method on the CIFAR10 dataset) and ARD-VAE on the CIFAR10 dataset with samples generated using all the latent dimensions. The FID scores of the RAE increase with the increase in the size of the latent space. Whereas, the FID scores of the ARD-VAE is consistent.

is determined using cross-validation on the reconstruction loss. Therefore, we have to retrain the DLVM multiple times on a *new* dataset with additional hyperparameter tuning to determine the number of latent dimensions, as the complexity of the dataset is *unknown* to us. This is an unrealistic and tedious approach. In contrast, the ARD-VAE can start training with a reasonable estimate of $\beta \in [0.05, 0.5]$ (refer to Tab. 6 for the real datasets) and sufficiently large latent space to learn meaningful representations and model the data distribution.

In this experiment, we demonstrate the impact of an incorrectly chosen latent dimension, i.e., significantly bigger, on the performance of the RAE [4] that produces comparable results to the ARD-VAE under different evaluation scenarios studied in this work. From the results reported in Tab. 14, we observe that the performance of the RAE is strongly affected by the choice of the initial size of the latent space. In contrast, the ARD-VAE is not sensitive to the initial size of the latent space and produces consistent results with $\beta = 0.05$. The results in Tab. 14 illustrate the necessity of methods, such as the ARD-VAE, that can automatically identify the relevant latent dimensions required to

| Bottleneck size | CIFAR10 ($L = 128$) | | | | ImageNet ($L = 256$) | | | |
|---|---|---|---|---|---|---|---|---|
| | Initial | Active | Pruned Axes = Initial−Active | FID↓ | Initial | Active | Pruned Axes = Initial−Active | FID↓ |
| $L/2$ | 64 | $60.40 \pm 0.49$ | 3.60 | $104.03 \pm 1.33$ | 128 | $101.33 \pm 0.47$ | 26.67 | $129.63 \pm 1.26$ |
| $L$ | 128 | $105.80 \pm 1.33$ | 22.20 | $87.56 \pm 1.21$ | 256 | $152.00 \pm 1.10$ | 104.00 | $121.21 \pm 1.16$ |

Table 15. In this result, we address the issue when the initial size of latent space $L$ is smaller than the optimum size identified by the ARD-VAE. In this analysis, we study the FID scores of the generated samples and the number of Active dimensions identified with the reduced size of the latent space for the complex CIFAR10 and ImageNet datasets. The ARD-VAE *preserves* most of the latent dimensions for the reduced size of the latent space, i.e., $L/2$, relative to the bigger latent space, $L$, which shows the effectiveness of the proposed method. As expected, the number of Active dimensions is less under the setting of $L/2$, subsequently increasing the FID scores.

model the distribution of a real dataset.

In another experimental setting, we adopt a conservative approach and train the ARD-VAE with a reduced number of latent dimensions on the CIAFR10 and ImageNet datasets than the optimum dimensions determined by the ARD-VAE. For example, we train the ARD-VAE on the CIFAR10 dataset in a latent space of size $L/2 = 64$ when the optimum size determined determined by the ARD-VAE is $\sim 106$. The motivation of this experiment is to evaluate the impact on the metric scores produced and the number of relevant dimensions identified by the ARD-VAE under such settings. We keep the hyperparameter *unchanged* when using the reduced size of the latent space, such as $L/2$. From the results reported in the Tab. 15, we observe that the ARD-VAE prunes less dimensions when trained on a smaller latent space and as expected, the FID scores increase. However, the impact on the metric scores is not significant.

## References


[1] Chris Burgess and Hyunjik Kim. 3d shapes dataset. https://github.com/deepmind/3d-shapes/, 2018. 3, 4, 5, 6

[2] Cedric De Boom, Samuel Wauthier, Tim Verbelen, and Bart Dhoedt. Dynamic narrowing of vae bottlenecks using geco and l0 regularization. In *International Joint Conference on Neural Networks (IJCNN)*, 2021. 4, 5

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 6

[4] Partha Ghosh, Mehdi S. M. Sajjadi, Antonio Vergari, Michael Black, and Bernhard Scholköpf. From variational to deterministic autoencoders. In *International Conference on Learning Representations*, 2020. 4, 8

[5] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Conference on Neural Information Processing Systems*, 2017. 6

[6] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, 2018. 5

[7] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *International Conference on Learning Representations*, 2014. 1, 3

[8] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *International Conference on Learning Representations*, 2018. 5

[9] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, 2019. 5

[10] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017. 3, 5, 6

[11] Arnab Kumar Mondal, Sankalan Pal Chowdhury, Aravind Jayendran, Parag Singla, Himanshu Asnani, and Prathosh AP. Maskaae: Latent space optimization for adversarial auto-encoders. In *Uncertainty in Artificial Intelligence (UAI)*, 2020. 4

[12] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286, 2014. 1, 3

[13] Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lučić, Olivier Bousquet, and Sylvain Gelly. Assessing Generative Models via Precision and Recall. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 6

[14] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelköpf. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018. 4