A. Supplementary Materials

A.1. Additional reconstruction results

In Figures 2 and 8 we show a large collection of additional recontruction images on the CelebA-HQ [8] and LSUN Bedroom [14] datasets.

A.2. Smoothness of latent space

In this section we analyse the smoothness of the latent space learnt by DC-VAE. In Figure 5 we show additional high resolution (512×512) CelebA-HQ [8] images generated by an evenly spaced linear blending between two latent vectors. In Figure 7 of the main paper we show that DC-VAE is able to perform meaningful attribute editing on images while retaining the original identity. To perform image editing, we first need to compute the direction vector in the latent space that correspond to a desired attribute (e.g. has glasses, has blonde hair, is a woman, has facial hair). We compute these attribute direction vectors by selecting 20 images that have the attribute and 20 images that do not have the attribute, obtaining the corresponding pairs of 20 latent vectors, and calculating the difference of the mean. The results in Figure 7 of the main paper show that these direction vectors can be added to a latent vector to add a diverse combination of desired image attributes while retaining the original identity of the individual.

A.3. Effect of negative samples

In this section we analyse the effect of varying the number of negative samples used for contrastive learning. Figure 1 shows the reconstruction error on the CIFAR-10 [9] test set as the negative samples is varied. We observe that a higher number of negative samples results in better reconstruction. We choose 8096 for all of our experiments because of memory constraints.

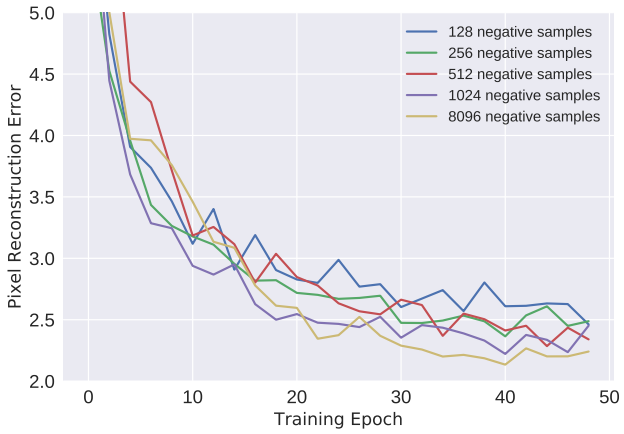


Figure 1: Pixel reconstruction error on CIFAR-10 [9] test set for varying number of negative samples

A.4. Dataset details

CIFAR-10 comprises 50,000 training images and 10,000 test images with a spatial resolution of 32×32. STL-10 is a similar dataset that contains 5,000 training images and 100,000 unlabeled images at 96×96 resolution. We follow the procedure in AutoGAN [4] and resize the STL-10 images to 32×32. The CelebA dataset has 162,770 training images and 19,962 testing images, CelebA-HQ contains 30,000 images of size 1024×1024, and LSUN Bedroom has approximately 3M images. For CelebA-HQ we split the dataset into 29,000 training images and 1,000 validation images following the method in [6]. We resize all images progressively in these three datasets from (4×4) to (512×512) for the progressive training.

A.5. Network architecture diagrams

In Figure 9 we show the detailed network architecture of DC-VAE for input resolutions of 32×32. Note that the comparison results shown in Figure 3 and Table 1 of the main paper, for VAE, VAE/GAN, VAE w/o GAN, and our proposed DC-VAE are all based on the same network architecture (shown in Figure 9 here), for a fair comparison.

The network architectures shown in Figure 9 are adapted closely from the networks discovered by [4] through Neural Architecture Search. The DC-VAE developed in our paper is not tied to any particular CNN architecture. We choose the AutoGAN architecture [4] to start with a strong baseline. The decoder in Figure 9 matches the generator in [4]. The encoder is built by modifying the output shape of the final linear layer in the discriminator of AutoGAN [4] to match the latent dimension and adding spectral normalization. The discriminator is used both for classifying real/fake images, and contrastive learning. For each layer we choose, we first apply 1x1 convolution and a linear layer, and then use this feature as an input to the contrastive module. For experiments at 32×32, we pick two different positions: the output of second residual conv block (lower level) and the output of the first linear layer (higher level). For experiments on higher resolution datasets we use a Progressive GAN [8] Generator and Discriminator as our backbone and apply similar modifications as described above.

A.6. Further details about the representation learning experiments

As seen in Table 6 of the main paper, we show the representation capability of DC-VAE following the procedure outlined in [2]. We train our model on the MNIST dataset [10] and measure the transferability though a classification task on the latent embedding vector. Specifically, we first pretrain the DC-VAE model on the training split of the MNIST dataset. Following that we freeze the DC-VAE model and train a linear classifier that takes latent embedding vector as the input and predicts the class label of the original image.

Figure 2: Additional CelebA-HQ [8] reconstruction images (resolution 512×512) generated by DC-VAE (ours)

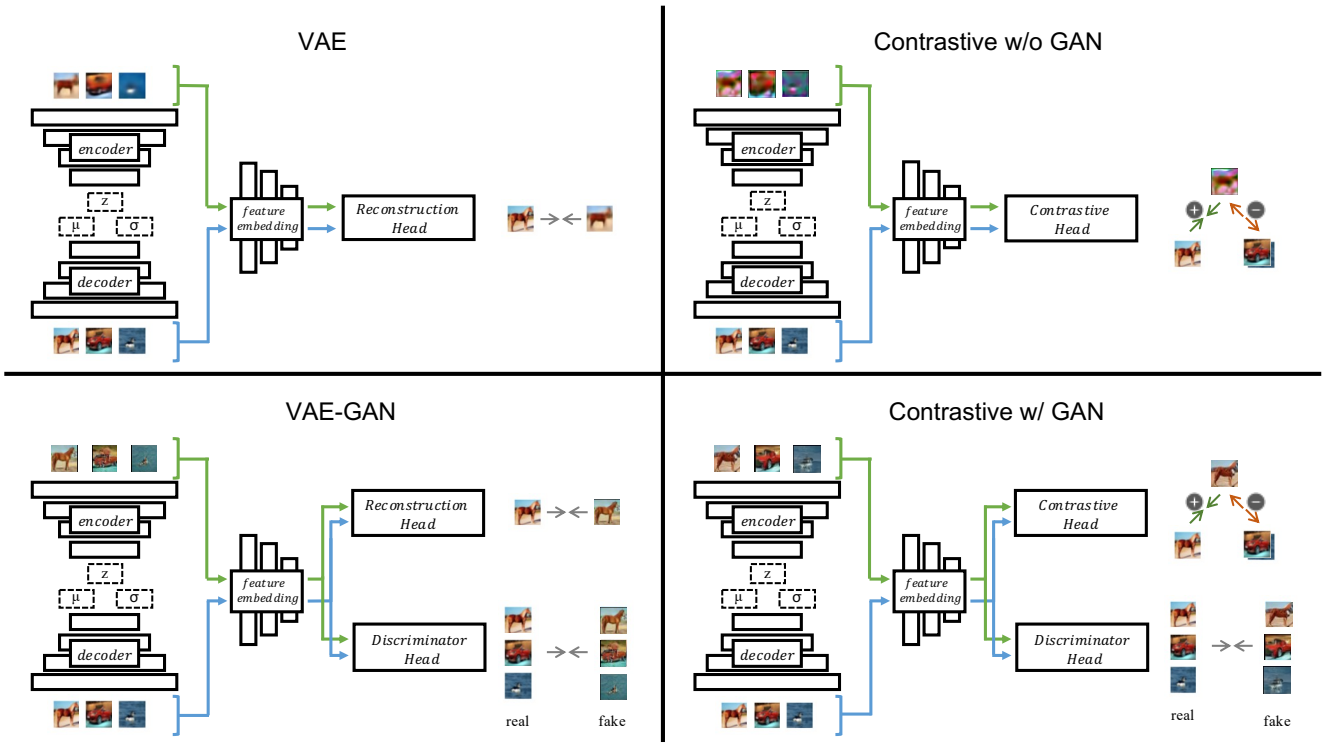
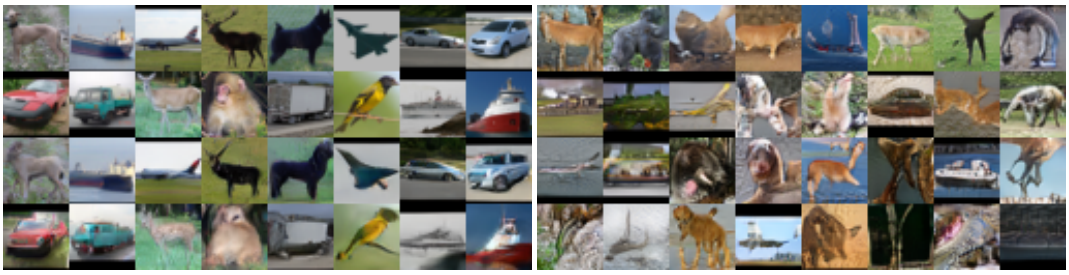


Figure 3: Visualization of the effect of adding each instance level and set level objectives. Table 1 and Figure 3 (in the main paper) contain FID [5] results and qualitative comparisons on the CIFAR-10 [9] that correspond to these settings.



(a) STL-10 Reconstructions generated by DC-VAE (b) STL-10 Samples generated by DC-VAE

Figure 4: DC-VAE reconstruction (a) and synthesis results (b) on STL-10 [1] images (resolution 32×32). In (a) the top two rows are input images and the bottom two rows are the corresponding reconstruction images.

A.7. Evaluation details

In Tables 1 and 8 of the main paper the perceptual distance is computed as the average MSE distance of the features extracted by a pretrained VGG-16 network. We borrow from [7] and use the activation of the relu4_3 layer. For computing the FID scores we follow the standard practice ([6], [11]) and use 50,000 generated images. In Table 4 of the main paper we use the 256×256 version of DC-VAE model trained on CelebA-HQ [8] for a fair comparison with other methods which are trained at the same resolution.

Table 1: Comparison on CIFAR-10 with a DCGAN [12] backbone. *Code and saved models are not provided for this method.

Method	FID Sampling↓	Perceptual Distance ↓
ALI / BiGAN	86.37	98.47
VEEGAN*	95.2	-
DC-VAE	78.06	80.99

Table 2: Comparison on CIFAR-10 with AutoGAN [4] backbone (all methods use multi-scale learning with the same intermediate layers). Autoencoder baseline is trained to minimize L2 loss on a pretrained VGG16 feature space.

Method	**FID Sampling**↓	**Perceptual Distance**↓
Autoencoder (without KL)	N/A	40.2
GAN	14.2	N/A
VAE/GAN (L2; feature space)	39.8	57.2
VAE/GAN (L1; feature space)	34.8	93.1
VAE/GAN (L2; pixel space)	33.4	63.4
VAE/GAN (L1; pixel space)	29.5	57.7
DC-VAE (ours)	17.9	52.9

A.8. Further comparison on CIFAR-10

We also trained VAE-GAN with L1/L2 loss in both pixel and feature spaces. We discover that our proposed model is consistently better than these baselines. Results are shown in Table 2.

A.9. Training with a weaker backbone

To make a fair comparison with [3] and [13], we trained our model using [12]. Results are shown in table 1. We discover that even when we control the capability of our model backbone, our method improves both FID score and perceptual distance.

Figure 5: Additional latent space interpolations on CelebA-HQ [8] (resolution 512×512)

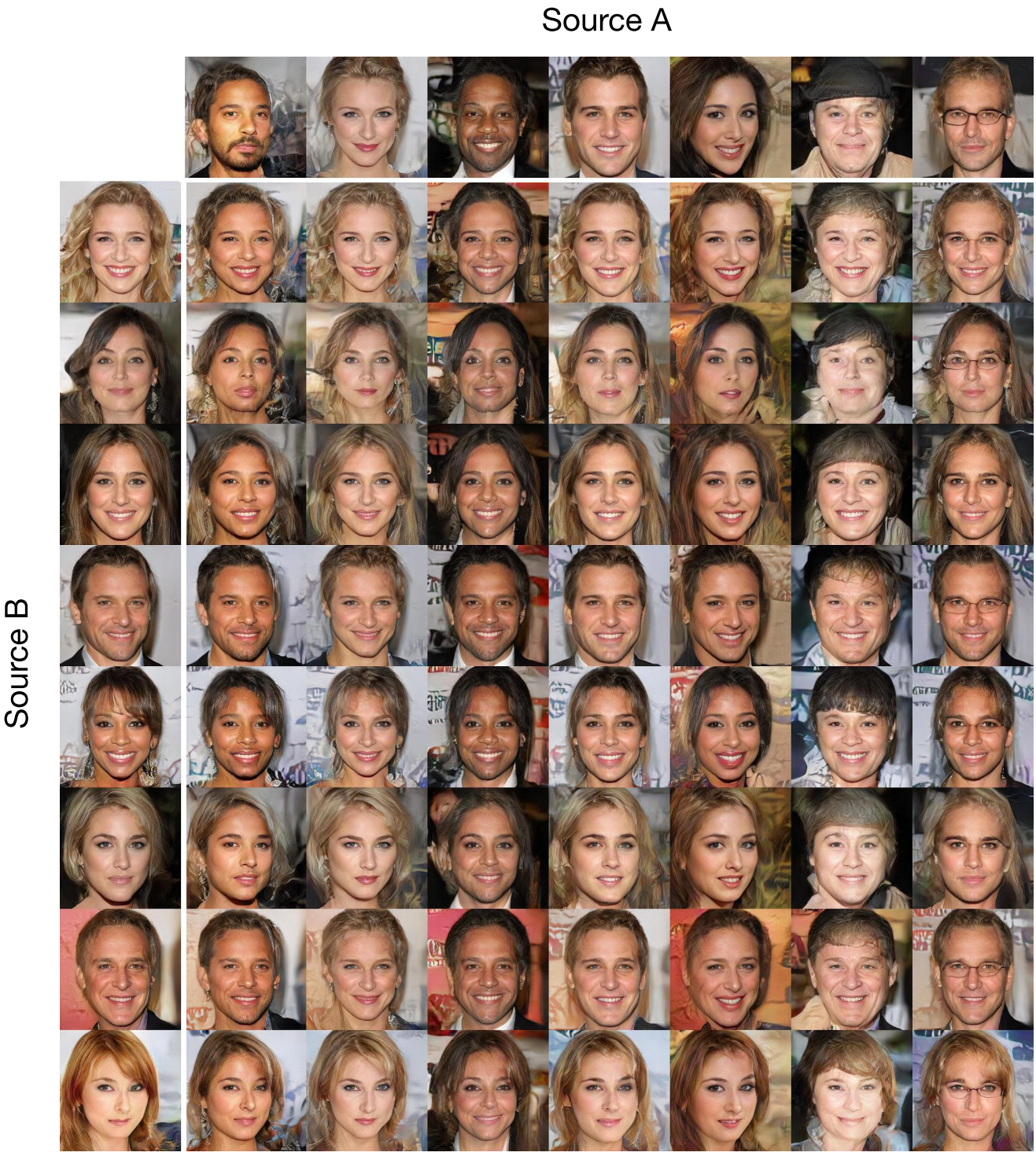


Figure 6: Latent Mixing results on CelebA-HQ [8]. Each combined image in the grid is generated by replacing an arbitrary subset of Source A latent with the corresponding Source B latent.

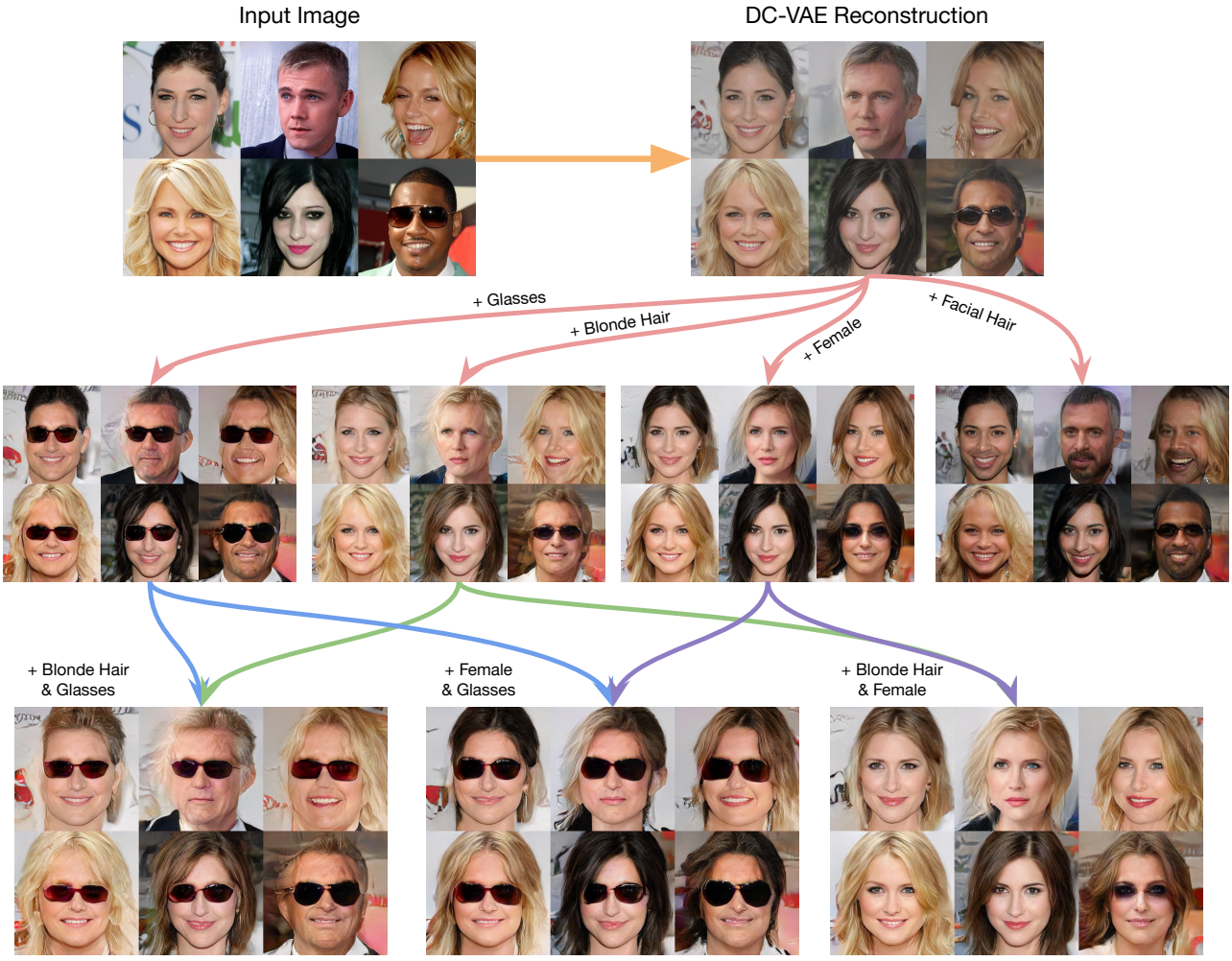


Figure 7: Additional image editing on CelebA-HQ [8] reconstruction images (resolution 512×512)

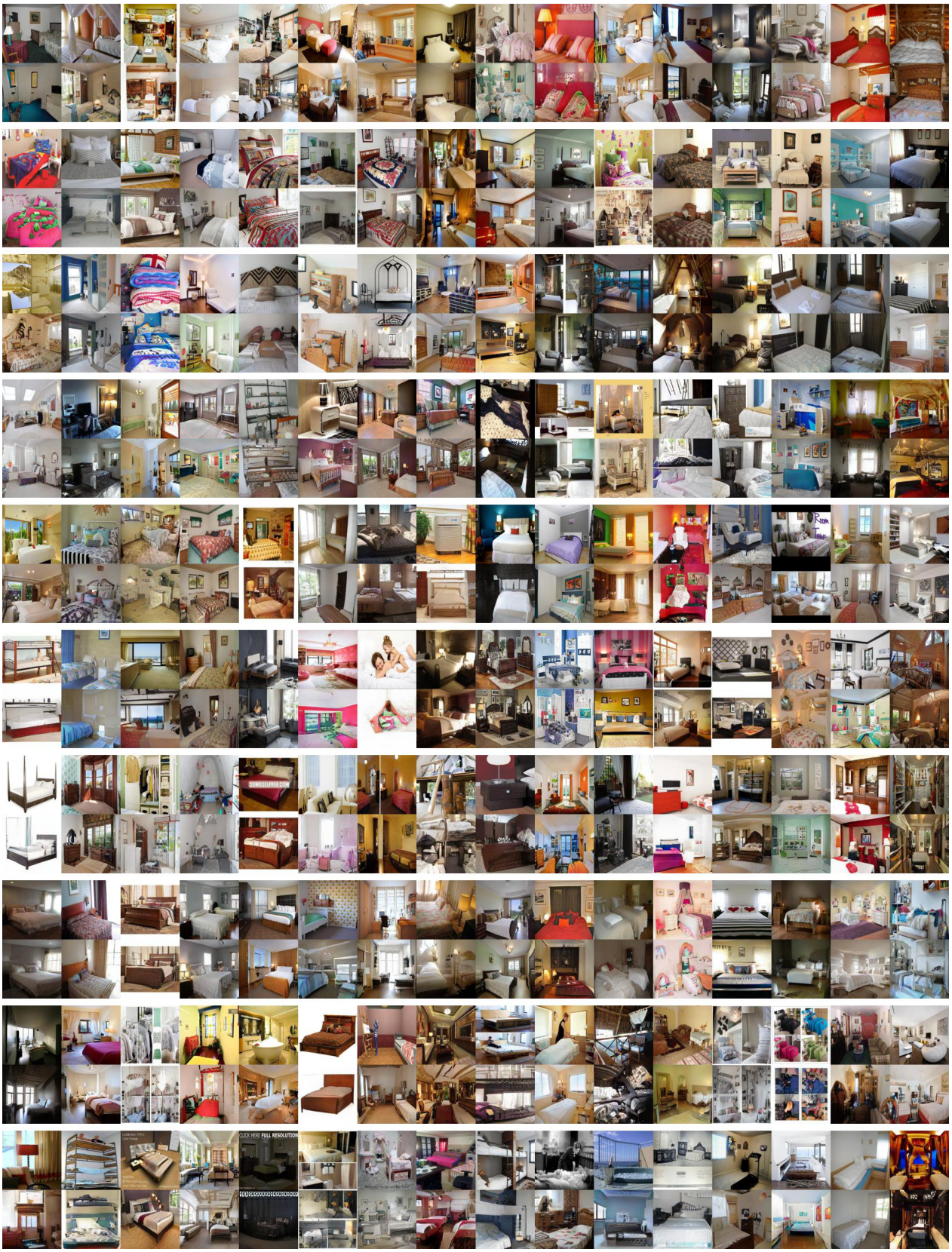
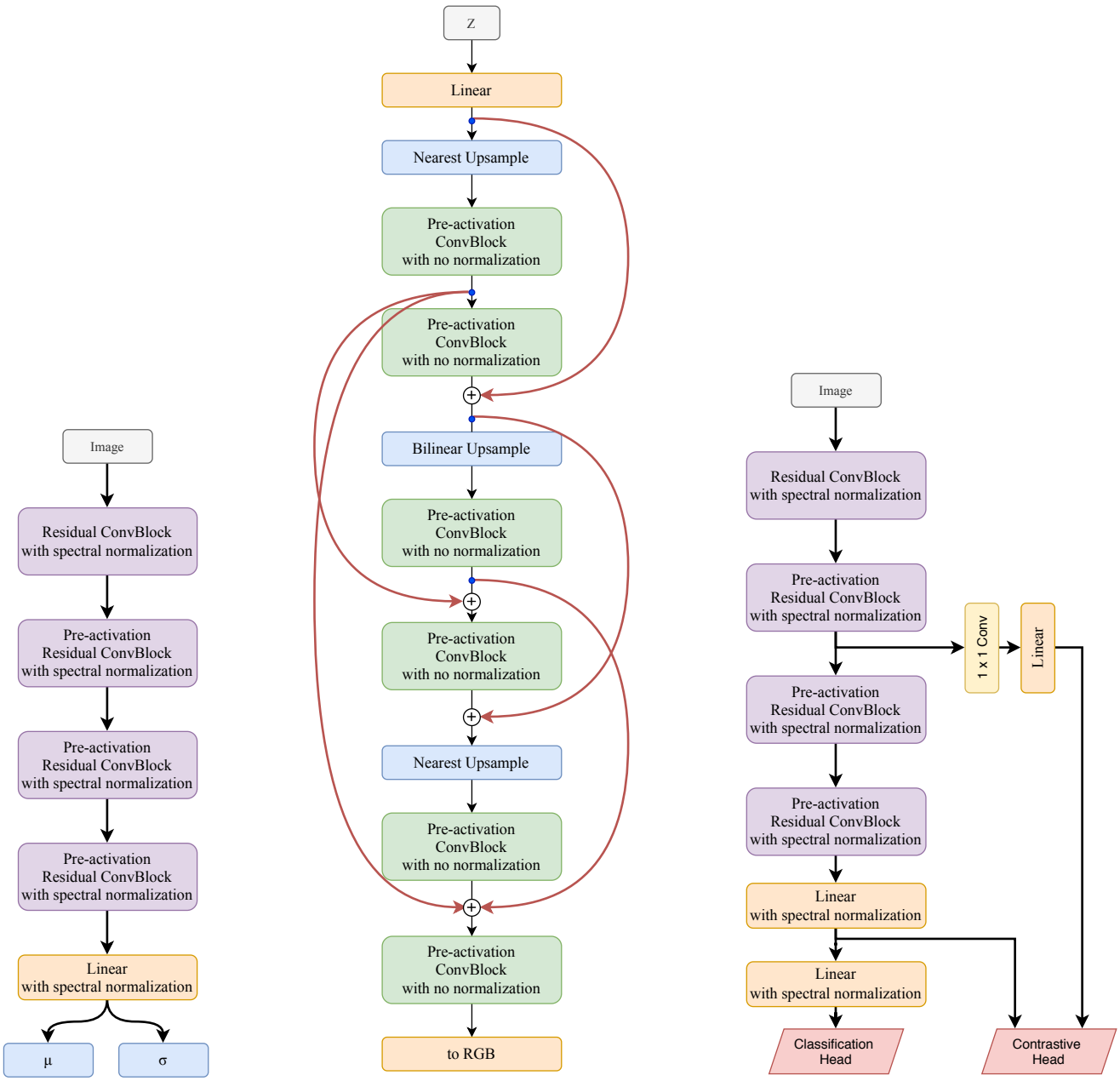


Figure 8: Additional LSUN Bedroom [14] reconstruction images (resolution 128×128)



(a) Encoder (b) Decoder (c) Discriminator

Figure 9: Network architecture of DC-VAE for resolution 32×32 for CIFAR-10 [9] and STL-10 [1]. (a) is the Encoder. (b) is the Decoder. (c) is the Discriminator.

References

[1] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, pages 215–223, 2011.

[2] Zheng Ding, Yifan Xu, Weijian Xu, Gaurav Parmar, Yang Yang, Max Welling, and Zhuowen Tu. Guided variational autoencoder for disentanglement learning. In *CVPR*, 2020.

[3] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. In *ICLR*, 2017.

[4] Xinyu Gong, Shiyu Chang, Yifan Jiang, and Zhangyang Wang. Autogan: Neural architecture search for generative adversarial networks. In *ICCV*, 2019.

[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 *Advances in Neural Information Processing Systems*, 2017.

[6] Huaibo Huang, Ran He, Zhenan Sun, Tieniu Tan, et al. Introvae: Introspective variational autoencoders for photographic image synthesis. In *Advances in Neural Information Processing Systems*, pages 52–63, 2018.

[7] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[8] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018.

[9] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[10] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[11] Stanislav Pidhorskyi, Donald Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders, 2020.

[12] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

[13] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems*, pages 3308–3318, 2017.

[14] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.