Supplementary Materials: Nested Scale-Editing for Conditional Image Synthesis

In this section, we discuss the implementation details and demonstrate more qualitative results of our experiments on multimodal image outpainting, image super-resolution, cross-domain image translation, and text-to-image translation.

A. Image Outpainting

The detailed implementation of our decoder network architecture for image inpainting is shown in Fig.3 in the main paper. At each spatial scale, the number of channels for feature activation is 512. The conditional code is the feature vector of the occluded image encoded by a standard encoder network. The detailed implementations of the encoder network are listed in Table 1. We use negative slope of 0.2 for all LeakyReLU layers throughout the network. We employ the following abbreviation: N = Number of filters, K = Kernel size, S = Stride, P = Padding. ”Conv” and ”SN” denote convolutional layer and instance normalization respectively.

Layer	**Hyper-parameters**
1	Conv(N64-K4-S2-P1) + LeakyReLU
2	Conv(N128-K4-S2-P1) + IN + LeakyReLU
3	Conv(N256-K4-S2-P1) + IN + LeakyReLU
4	Conv(N512-K4-S2-P1) + IN + LeakyReLU
5	Conv(N256-K4-S2-P1) + IN + LeakyReLU
6	Conv(N256-K4-S2-P1) + IN + LeakyReLU
7	Conv(N128-K1-S2-P1) + LeakyReLU

Table 1: Encoder network for image outpainting and super-resolution.

The weights for the adversarial loss, disentangle loss, and diversity loss are all set to be ones. To enforce the diversity of synthesis, we sample $N = 4$ random variables at each iteration. We set the relaxation hyperparemeter α in the diversity hinge loss to be 0.8. With batch size of 24, we train the network using Adam optimizer [3] with learning rate of 2e-4, beta1 of 0.5, beta2 of 0.999.

B. Image Super-Resolution

Our super-resolution network is mostly similar to the network used for image outpainting with two major differences. The first difference is that we do not decode any image lower than the low-resolution scale (16x16), since there is no need to edit visual details below the input resolution scales. Thus, our decoder starts to generate images at scale of 32x32 and enforces the downsampled sample of the 32x32 images to be the same as the ground-truth 16x16 low-resolution image. The disentanglement loss for scales of 64 and 128 are the same as the outpainting newtork. In addition, we add skip connections from the encoder to the decoder for the purpose of preserving low-resolution structural information. The encoder for the low-resolution image is the same as the encoder used in image outpainting, which is shown in Table 1. We also use the same optimizer and hyperparameters for both the image super-resolution and image outpainting.

C. Cross-Domain Translation

For the cross-domain translation task we adapted the MUNIT network[2]. In terms of network architecture, we use exactly the same content and style encoders as the original and we only modify the decoder, where we add an additional convolution for image output at the 128x128 resolution, and correspondingly the discriminator for it. We used the default multi-resolution discriminator as in the original implementation. For details of the architecture we refer reader to [2] and its official github repository [1]. In terms of losses, in addition to the original reconstruction losses and discriminator losses, we calculated the proposed disentangle loss L_{disent} between the two levels as well as the normalized diversity loss [4] on each level. We use the following weights for losses: weight of adversarial loss $\lambda_{GAN} = 1$; weight of image reconstruction loss $\lambda_{xw} = 10$; weight of style reconstruction loss $\lambda_{sw} = 1$; weight of image reconstruction loss $\lambda_{cw} = 1$; weight of normalized diversity loss $\lambda_{ndiv} = 1$; weight of the disentangle loss $\lambda_{disent} = 1$. An illustration of the network architecture as well as the added losses is shown in Fig.2. We optimize the network using an Adam optimizer with learning rate of $1e-4$, $beta1$ of 0.5 and $beta2$ of 0.999 with batch size of 2.

D. Text-to-Image synthesis

Other than the image outpainting, image superresolution and cross-domain translation, we also evaluate our proposed multi-scale disentangle loss and the normalized diversity loss on the task of text-to-image synthesis. Our implementation is based on the StackGAN++ [8]. We refer interested reader to [2] for the original implementation. We use pretrain text embedding from [6], as in [8] and [5]. We keep the original text embedding sampling unchanged but incorporate two changes within the decoder. First, we incorporate the adaIn layer [1] for each refine stage, which allows injection of random latent vector at each stage. In comparison the original implementation only inject latent random vector at the init stage. Second, we added two more level of image output to the original image. The original Stack-

[1] https://github.com/NVlabs/MUNIT.

[2] https://github.com/hanzhanggit/StackGAN-v2

Method	Quality ↓	Diversity ↑
MSGAN[5]	**18.64**	0.661
Ours	20.88	**0.668**

Table 2: Quantitative comparison with state-of-the-art approaches on the cross-modal image-to-image translation task.

GAN network outputs 64/128/256 images. We extended the network to output 16/32 images. With the two changes in place, we add the proposed disentangle loss and normalized diversification loss to it. Detailed architecture of the modified StackGAN++ is illustrated in Fig.3. We tested our network on the cub_200_2011 [7] birds dataset. As in the image outpainting, image superresolution and cross-domain translation task, we achieve scale-specific editing by injecting different latent codes at each scale at test time, as shown on 7. Quantitatively, our network achieved similar image quality (measured by FID) and slightly higher diversity (measured by LPIPS) as previous state-of-the-art from [5], as shown by Table.2. We optimize the network using an Adam optimizer with learning rate of $2e-4$, $beta1$ of 0.5 and $beta2$ of 0.999 with a batch size of 4.

References

[1] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017. 1

[2] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 172–189, 2018. 1, 3

[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[4] Shaohui Liu, Xiao Zhang, Jianqiao Wangni, and Jianbo Shi. Normalized diversification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10306–10315, 2019. 1

[5] Qi Mao, Hsin-Ying Lee, Hung-Yu Tseng, Siwei Ma, and Ming-Hsuan Yang. Mode seeking generative adversarial networks for diverse image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1429–1437, 2019. 1, 2

[6] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016. 1

[7] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 2

[8] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1947–1962, 2018. 1, 3

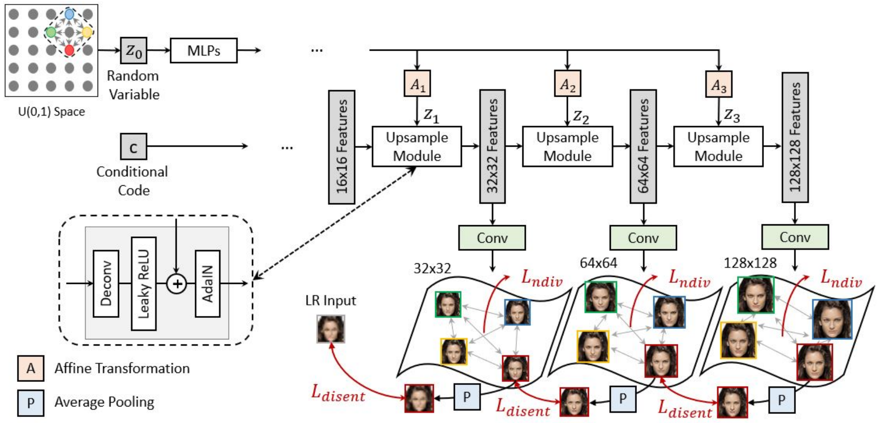

Figure 1: The model architecture of super-resolution decoder network.

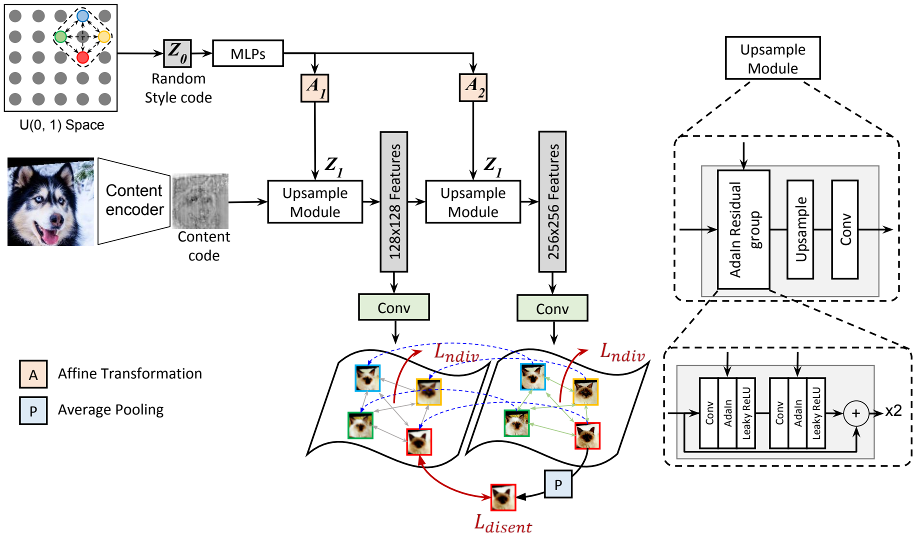

Figure 2: The model architecture of modified MUNIT [2] decoder network.

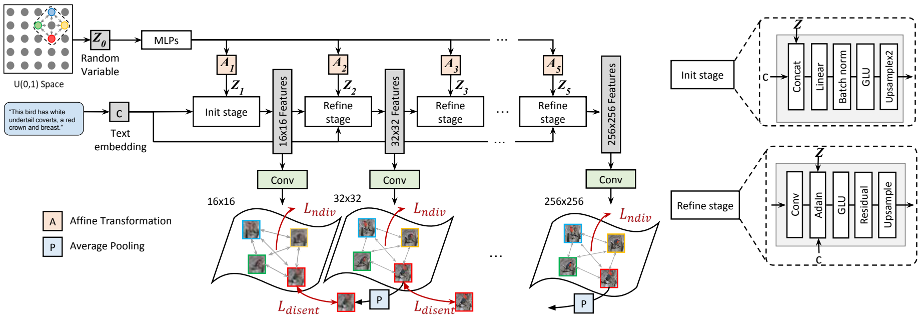

Figure 3: The model architecture of modified StackGAN++ [8] decoder network.

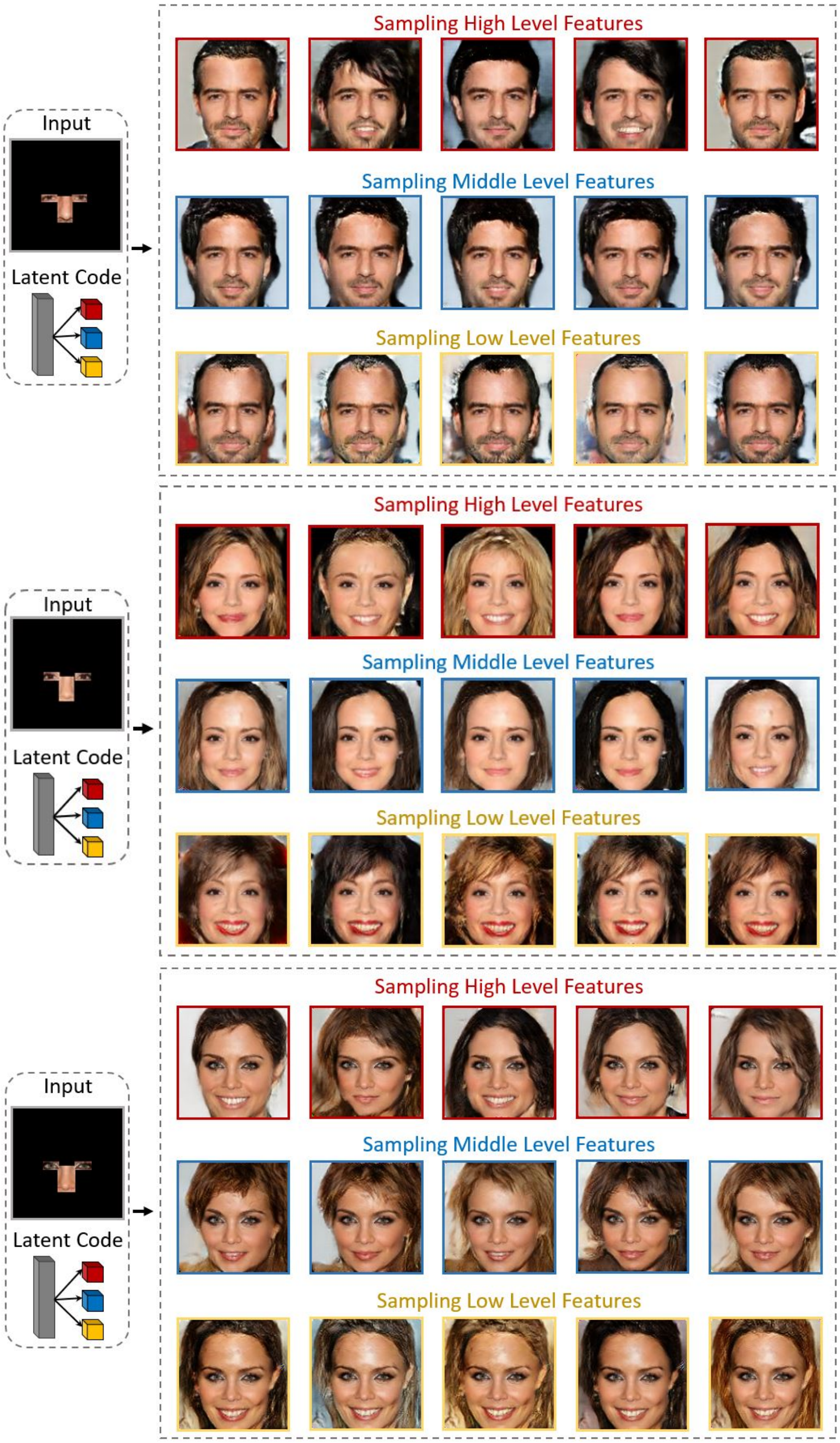
Sampling High Level Features
Input
Latent Code
Sampling Middle Level Features
Sampling Low Level Features
Sampling High Level Features
Input
Latent Code
Sampling Middle Level Features
Sampling Low Level Features
Sampling High Level Features
Input
Latent Code
Sampling Middle Level Features
Sampling Low Level Features

Figure 4: Qualitative results for scale-editing for image outpainting. We vary random variables at scale of 4 to edit high-level features, vary random variables at scale of 8 and 16 to edit the middle-level features, and vary random variables at scales of 32, 64, 128 to edit the low-level features.

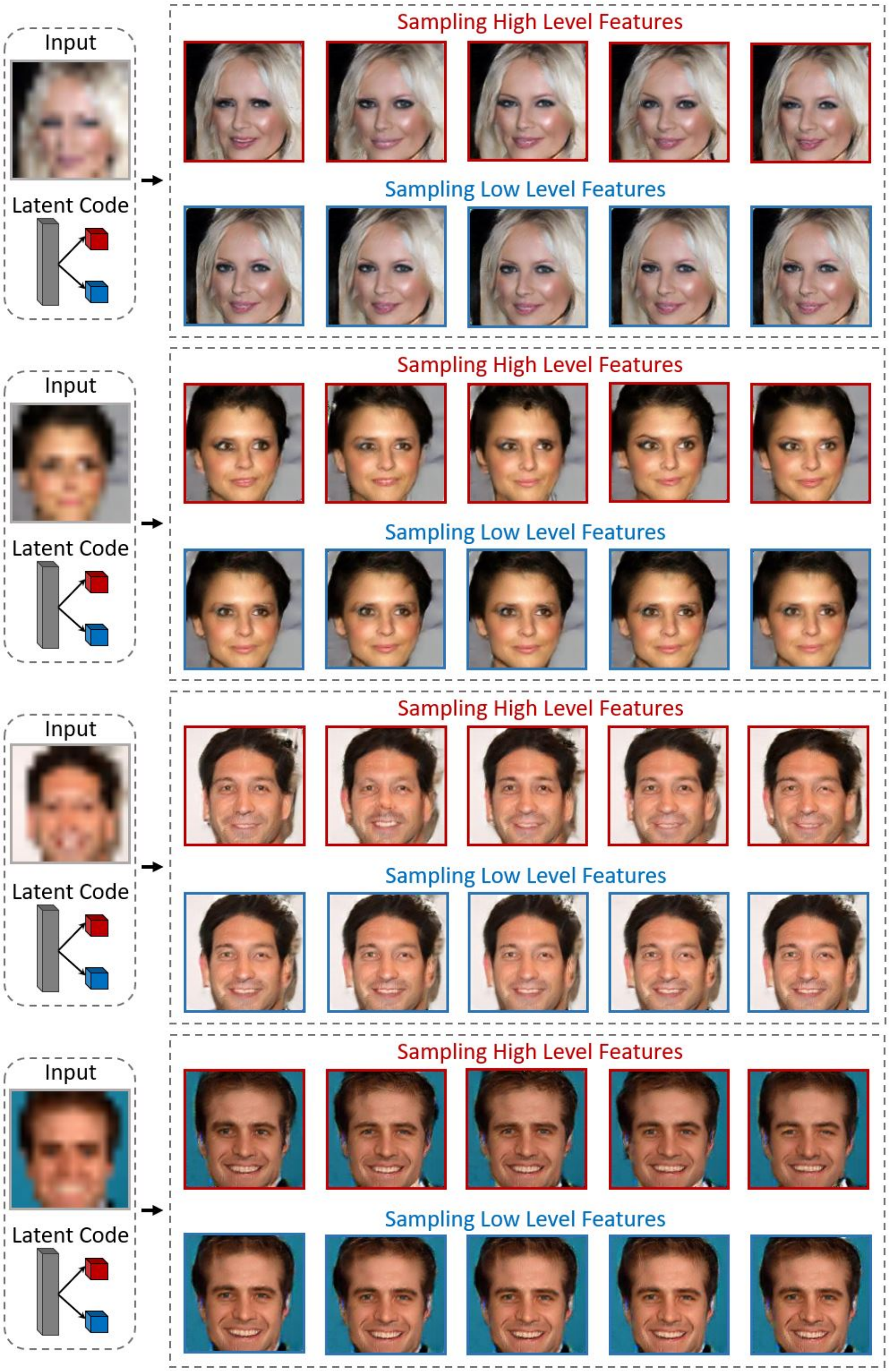
Input
Latent Code
Sampling High Level Features
Sampling Low Level Features
Input
Latent Code
Sampling High Level Features
Sampling Low Level Features
Input
Latent Code
Sampling High Level Features
Sampling Low Level Features
Input
Latent Code
Sampling High Level Features
Sampling Low Level Features

Figure 5: Qualitative results for scale-editing for image super-resolution. We vary random variables at scale of 32 to edit high-level features, and vary random variables at scales of 64 and 128 to edit the low-level features. Note that the variations for this task are small in nature, and the low-level features in this super-resolution task only affect subtle textures. In the super-resolution task, our main goal is to generate multimodal outputs while preserving identities.

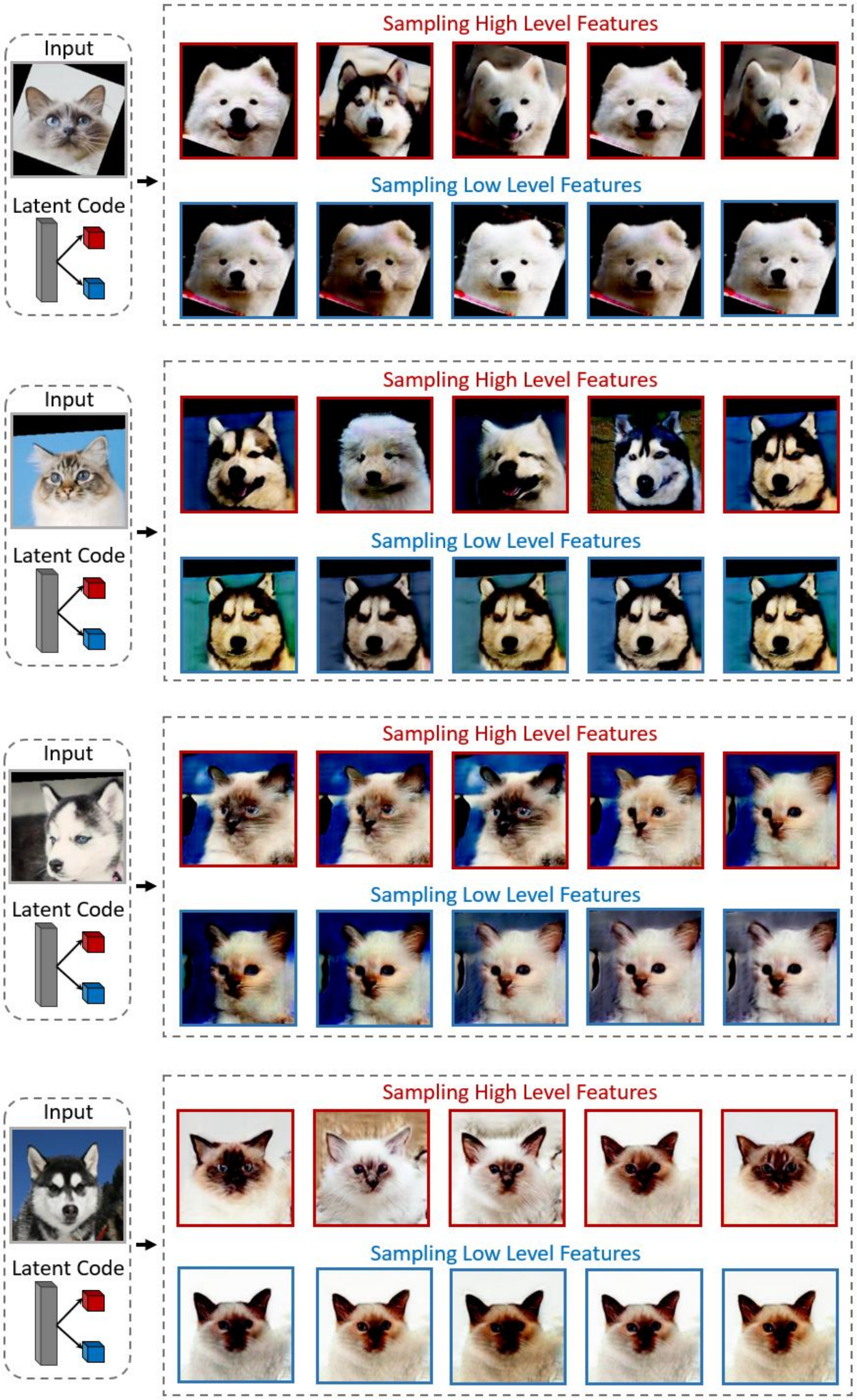
Input
Latent Code
Sampling High Level Features
Sampling Low Level Features
Input
Latent Code
Sampling High Level Features
Sampling Low Level Features
Input
Latent Code
Sampling High Level Features
Sampling Low Level Features
Input
Latent Code
Sampling High Level Features
Sampling Low Level Features

Figure 6: Qualitative results for scale-editing for cat2dog and dog2cat translation. Note that there are few modes of variation compared to other tasks, due to 1). primarily a small dataset size (871 cat and 1364 dog images) and 2). there are only two types of dogs (husky and samoyed) and mostly one type of cat (siamese) in the dataset.

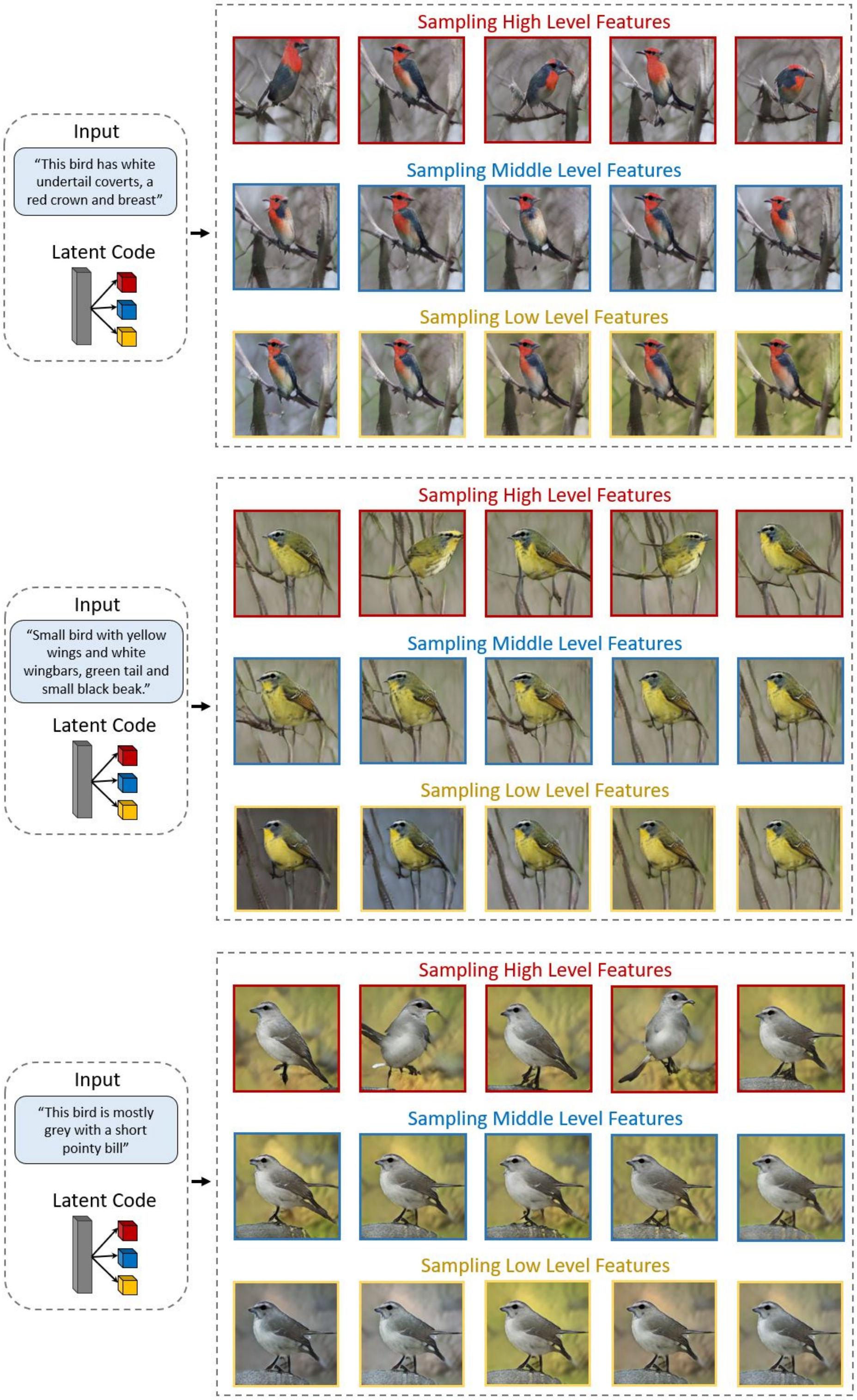
Input
"This bird has white undertail coverts, a red crown and breast"
Latent Code
Sampling High Level Features
Sampling Middle Level Features
Sampling Low Level Features
Input
"Small bird with yellow wings and white wingbars, green tail and small black beak."
Latent Code
Sampling High Level Features
Sampling Middle Level Features
Sampling Low Level Features
Input
"This bird is mostly grey with a short pointy bill"
Latent Code
Sampling High Level Features
Sampling Middle Level Features
Sampling Low Level Features

Figure 7: Qualitative results for scale-editing for text-to-image translation.