Supplementary Material for CG-NeRF: Conditional Generative Neural Radiance Fields for 3D-aware Image Synthesis

Kyungmin Jo† Gyumin Shim† Sanghun Jung Soyoung Yang
Jaegul Choo
Korea Advanced Institute of Science and Technology (KAIST)
Daejeon, Korea
{bttkm, shimgyumin, shjung13, sy_yang, jchoo} @kaist.ac.kr

1. Overview

In this supplementary material, we present implementation details and additional experiments, which support the validity of our proposed method. We provide details of implementation in Sec. 2, including datasets, training details, and discriminator architecture. We also present the experiment on local manipulation and additional results of our method in Sec. 3

2. Implementation Details

2.1. Dataset

We evaluate our CG-NeRF on various datasets including CelebA-HQ [3], CUB-200 [9], and Cats [11]. Unlike generation NeRF tasks [1, 5, 7], we aim to synthesize condition-aware outputs. We split all the datasets into training-validation-test sets and evaluate our method on the test set only. CelebA-HQ contains 30,000 images, resulting in 24,183 train 2,993 validation, and 2,824 test images. Among 11,799 images belonging to 200 bird species in CUB-200, we use 8,444 images following the preprocessing in GRAF [7] to filter only images that have a full-body shape. To minimize the correlation between training and test datasets, we divide the image set into 6315 training, 1064 validation, 1065 test images with respect to bird species. The CATS dataset contains 6,444 images, split into 4,784 training, 868 validation, and 792 test images. Since we use Clip [6] as a feature extractor, the tokenizer in CLIP is applied to tokenize text. After tokenizing the text, we randomly select up to 25 tokens for a condition input for text augmentation during training [10, 8].

For the prior camera pose distribution, we tune camera parameters according to the datasets. For CelebA-HQ dataset, we sample camera poses from a Gaussian distribution, with a vertical standard deviation of 0.155 radians and a horizonal standard deviation of 0.3 radians, and with $0.5 \times \pi$ mean radians. For Cats dataset, we sample camera poses from a Gaussian distribution, with a vertical standard deviation of 0.4 radians and a horizonal standard deviation of 0.7 radians, and with $0.5 \times \pi$ mean radians. We sample camera poses for CUB-200 dataset uniformly with horizontal range (-π,π) and vertical range (0, 2.97). We sample 48 sample per ray. We use a pinhole camera with a field of view of 30° for CelebA-HQ and CUB-200, and 12° for Cats dataset.

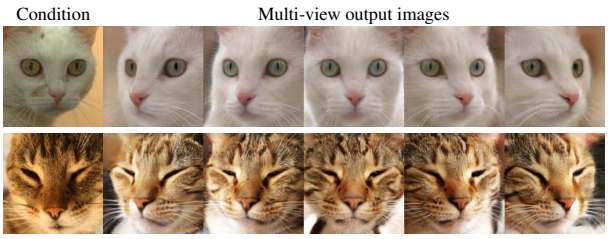
2.2. Training Details

The network is trained with an initial learning rate of 0.0004 using the ADAM optimizer with β_1= 0.9 and β_2= 0.999. We use PyTorch to implement the model on an NVIDIA 3090Ti GPU. For training the network with the pose-consistent diversity loss, we initially pre-train the network only with the adversarial loss and pose reconstruction loss for a certain steps, after which we fine-tune the network with the pose-consistent diversity loss in addition to the training objective. For the diversity-sensitive loss $\mathcal{L}_{\text{div}}$, weight λ_{div} is set to 0.5. Since the scale of pose reconstruction error varies for different datasets, the weight λ_{pose} commonly used for both the pose-penalty and pose reconstruction loss is set to 30, 20, and 2, respectively, for CelebA-HQ, Cats, and CUB-200 datasets. We train the network for 100,000 iterations with a batch size of 16 and adopt a moving average of parameters to evaluate our method.

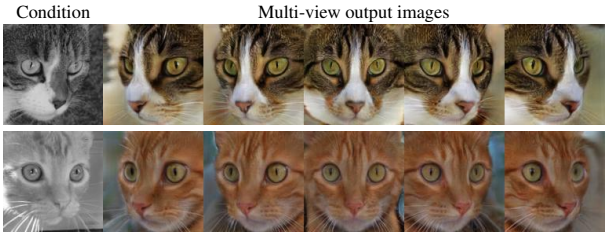
The values of hyper-parameters in Table. 1 are as follows. Since we apply CLIP as a feature extractor, the dimension L_c of global feature vector $\mathbf{c}$ is set to 512. The dimensions L_s and L_a of noise code $\mathbf{z}^s$ and $\mathbf{z}^a$ are both set to 100. Both the dimensions, L_γ and L_β, of frequencies γ and phase shifts β in each layer are 256. Also, the dimensions L_f of both feature vector $\mathbf{f}_{pj}$ and rendered feature $\mathbf{F}_p$ are set to 256, respectively.

The number of layers N^s in the shape block Φ^s is 5, and

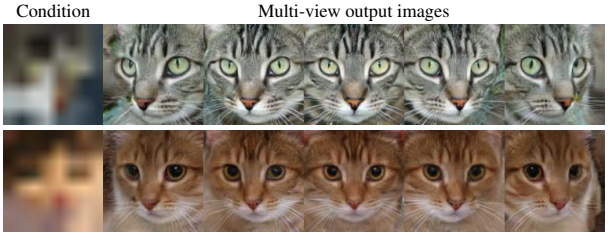
† Both authors contributed equally to this research.



(A) Conditioned on color images



(B) Conditioned on grayscale



(C) Conditioned on low-resolution

Figure 1: Multi-view output images on Cats dataset.

the number of layers N^a in the shape block Φ^a is 4. As in the previous generative NeRF models [7, 5], a style of shape is given to the block close to the input, and the style of appearance is provided to the block close to the output. In SIREN-based architecture among the previous models, we find that the appearance starts to change from earlier layers rather than the last layer (Fig. 2), so the input location of the appearance style can be adjusted.

The decoder network for upsampling feature map $\mathbf{F}$ consists of three residual blocks [2]. In each of the first two blocks, a nearest neighbor upsampling is applied to the output feature with a ratio of 2, which means the rendered 2D feature map $\mathbf{F} \in \mathbb{R}^{H_V \times W_V \times L_f}$ is upsampled two times to generate an output image $\hat{I} \in \mathbb{R}^{H \times W \times 3}$ (*e.g.,* $H_V \times W_V$ is 32×32 and $H \times W$ is 128×128).

2.3. Discriminator Architecture

Table 1 shows the architecture of the discriminator. Inspired by the recent work [1], we design the discriminator utilizing CoordConv layers [4] and a residual connection [2]. Since we train the conditional GAN that learns to identify the relevance between images and condition feature vectors, matching condition $\mathbf{e}$ is concatenated with the feature at 4×4 scale with spatial replication.

Figure 2: The role of each layer of the main architecture. We cumulatively replace frequency and phase shift in each MLP layer with the ones produced by other noise codes to observe the change of generated image according to the replacement (rows 1 and 4). Each slice in the bar above images in the first and fourth rows represents the 9 FiLM SIREN layers of our generator, and the color of the slice represents the source of the frequency and phase shift applied to each FiLM SIREN layer. Results replacing only frequency (rows 2 and 5) or phase shifts (rows 3 and 6) are visualized together to confirm their individual effects.

Type		Activation	Input Shape	Output Shape
Input	Input Image	-	3×128×128	3×128×128
Layers	Adapter Block (1×1)	LeakyReLU (0.2)	3×128×128	64×128×128
	Coord Conv (3×3)	LeakyReLU (0.2)	64×128×128	128×128×128
	Coord Conv (3×3)	LeakyReLU (0.2)	128×128×128	128×128×128
	AvgPooling	-	128×128×128	128×64×64
Layers	Coord Conv (3×3)	LeakyReLU (0.2)	128×64×64	256×64×64
	Coord Conv (3×3)	LeakyReLU (0.2)	256×64×64	256×64×64
	AvgPooling	-	256×64×64	256×32×32
Layers	Coord Conv (3×3)	LeakyReLU (0.2)	256×32×32	400×32×32
	Coord Conv (3×3)	LeakyReLU (0.2)	400×32×32	400×32×32
	AvgPooling	-	400×32×32	400×16×16
Layers	Coord Conv (3×3)	LeakyReLU (0.2)	400×16×16	400×16×16
	Coord Conv (3×3)	LeakyReLU (0.2)	400×16×16	400×16×16
	AvgPooling	-	400×16×16	400×8×8
Layers	Coord Conv (3×3)	LeakyReLU (0.2)	400×8×8	400×8×8
	Coord Conv (3×3)	LeakyReLU (0.2)	400×8×8	400×8×8
	AvgPooling	-	400×8×8	400×4×4
Input	Matching condition $\mathbf{e}$	-	712×4×4	712×4×4
Layers	Coord Conv (3×3)	LeakyReLU (0.2)	1112×4×4	400×4×4
	Coord Conv (3×3)	LeakyReLU (0.2)	400×4×4	400×4×4
	AvgPooling	-	400×4×4	400×2×2
Layers	Conv2d (2×2)	-	400×2×2	3×1×1

Table 1: The configuration of discriminator D_ψ (3 in Fig. 2). Matching condition $\mathbf{e}$ is concatenated with the feature at 4×4 scale with spatial replication.

3. Additional Experiment and Results

3.1. Local Manipulation

As shown in Fig. 3, our framework allows for a local editing, by manipulating specific local area (*e.g.,* eyes, mouth) of generated output images. To be specific, we replace a specific local region of an input condition with the one in other input conditions and visualize the generated results. In order to minimize the discontinuity of the input condition due to the boundary of the replaced region, we conduct the experiment using the sketch as an input con-

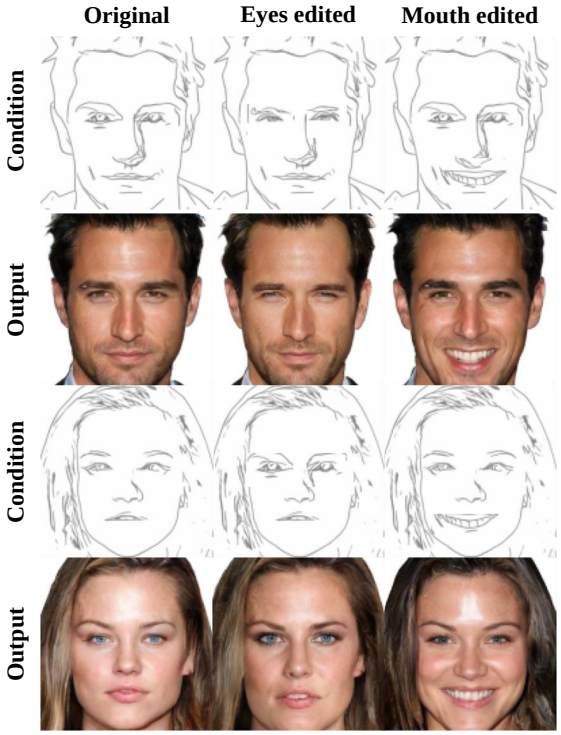


Figure 3: Results of local editing of sketch condition on CelebA-HQ dataset. To edit a specific area of the original sketch condition, we crop the desired face part (*e.g.,* eyes, mouth) from another sketch and paste it into the original sketch.

dition. As shown in the figure, our method is capable of faithfully reflecting the desired local area into the generated output image with negligible changes in unaltered region.

3.2. Additional Results of CG-NeRF

We include additional visual results to show image quality on different types of conditions. Fig. 1 shows multi-view output images on Cats dataset, which demonstrates the view consistency of our method on another dataset. Fig. 4 shows output images dependent on different condition types on CelebA-HQ. Each image set (a)-(f) in the figure contains five conditions (top row) obtained from the same image-text pair, respectively, and images (bottom row) generated from input conditions. Generally, images generated from a color image and a grayscale condition are shown to be similar due to rich information specified in the given condition. However, the image generated from the sketch has a similar shape to the two images but has a different color, so the color of hair, skin, or hat is changed. On the other hand, the image generated from low-resolution is similar in color to the two images but has a different shape, so a hat or glasses disappears and the gender changes. Finally, an image generated from text produces the most diverse images. This tendency also found in the Cats dataset (Fig. 5).

To avoid cherry picking, we visualize additional uncurated results on CelebA-HQ, CUB-200, and Cats dataset, which are selected with 20 random seeds, as shown in Fig. 6. The visualized images are selected from fake images existing in the manifold of the real image, which accounts for 89-91% of the test dataset for CelebA-HQ and Cats, and 82-84% for CUB-200 and FFHQ. Note that we do not manually select both input conditions and generated images.

References

[1] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proc. of IEEE conference on computer vision and pattern recognition (CVPR)*, pages 5799–5809, 2021.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. of IEEE conference on computer vision and pattern recognition (CVPR)*, pages 770–778, 2016.

[3] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[4] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. *arXiv preprint arXiv:1807.03247*, 2018.

[5] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proc. of IEEE conference on computer vision and pattern recognition (CVPR)*, pages 11453–11464, 2021.

[6] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[7] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *arXiv preprint arXiv:2007.02442*, 2020.

[8] Ming Tao, Hao Tang, Songsong Wu, Nicu Sebe, Xiao-Yuan Jing, Fei Wu, and Bingkun Bao. Df-gan: Deep fusion generative adversarial networks for text-to-image synthesis. *arXiv preprint arXiv:2008.05865*, 2020.

[9] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[10] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proc. of IEEE conference on computer vision and pattern recognition (CVPR)*, pages 1316–1324, 2018.

[11] Weiwei Zhang, Jian Sun, and Xiaoou Tang. Cat head detection-how to effectively exploit shape and texture features. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 802–816. Springer, 2008.

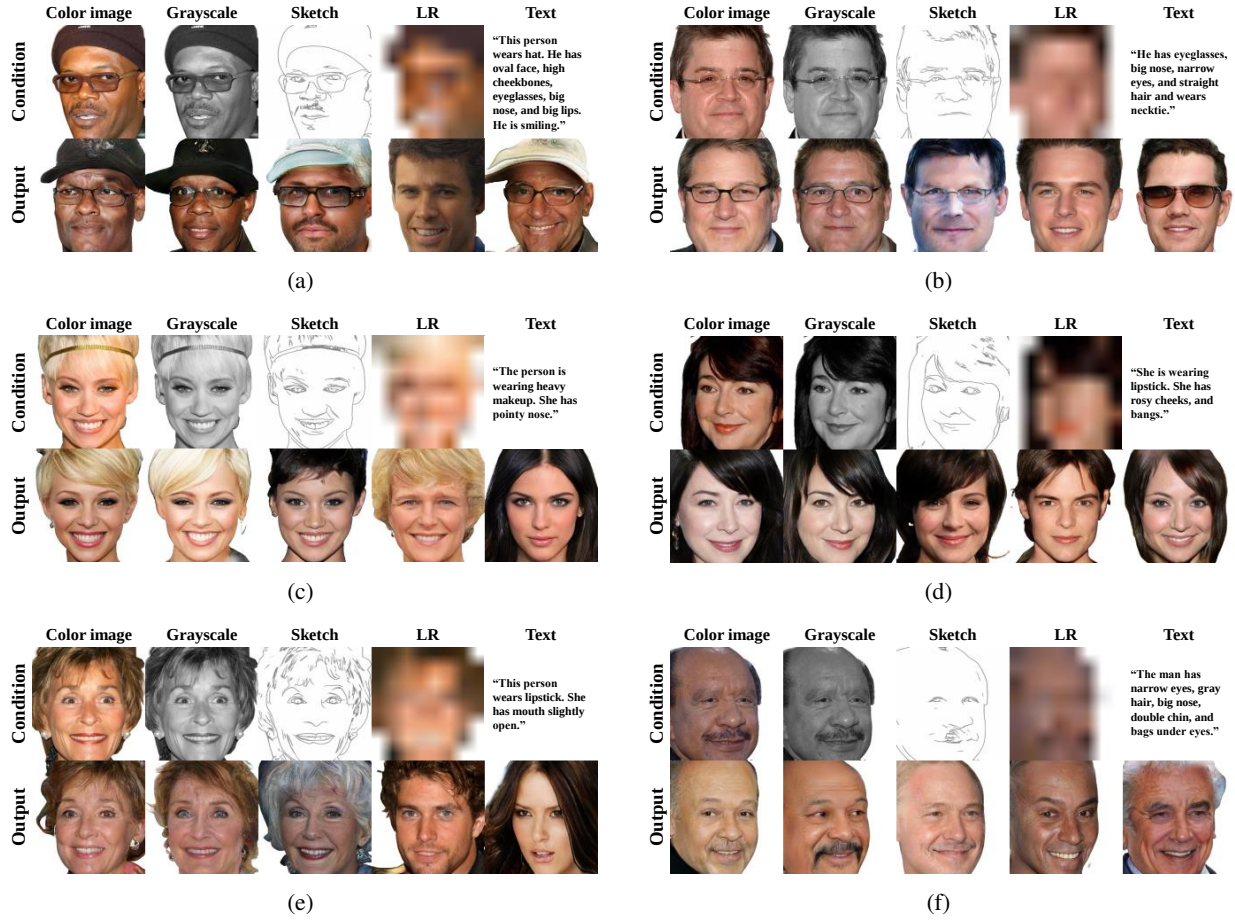


Figure 4: Output images dependent on different condition types on CelebA-HQ dataset.

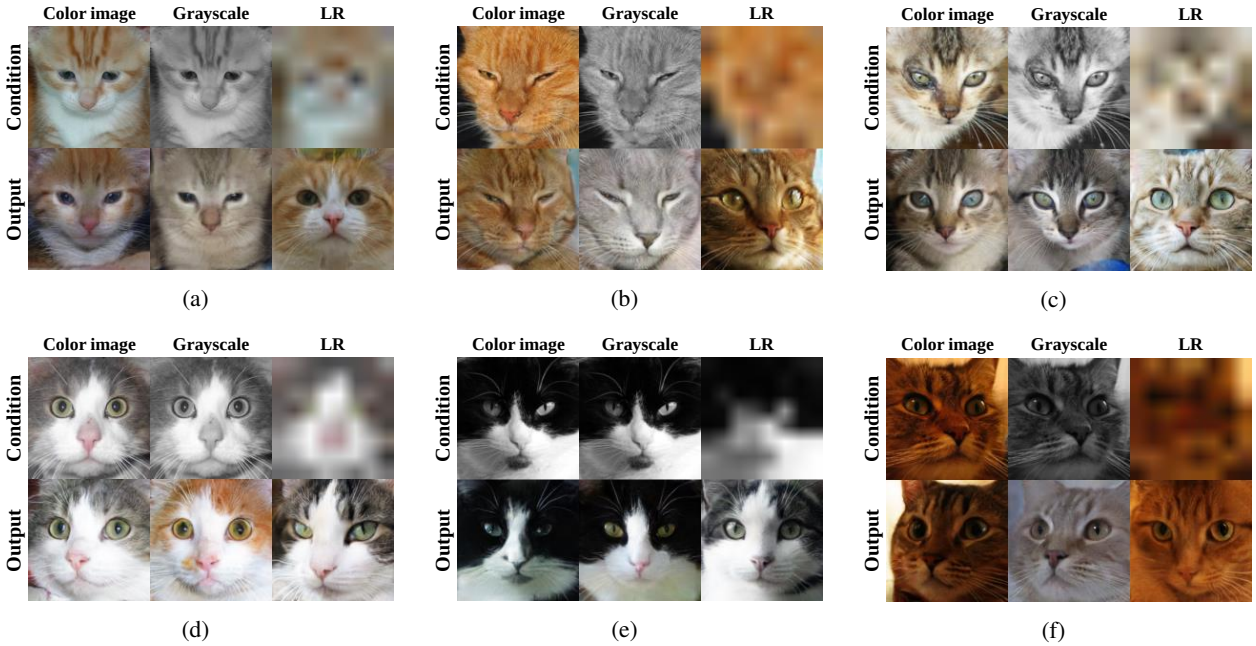
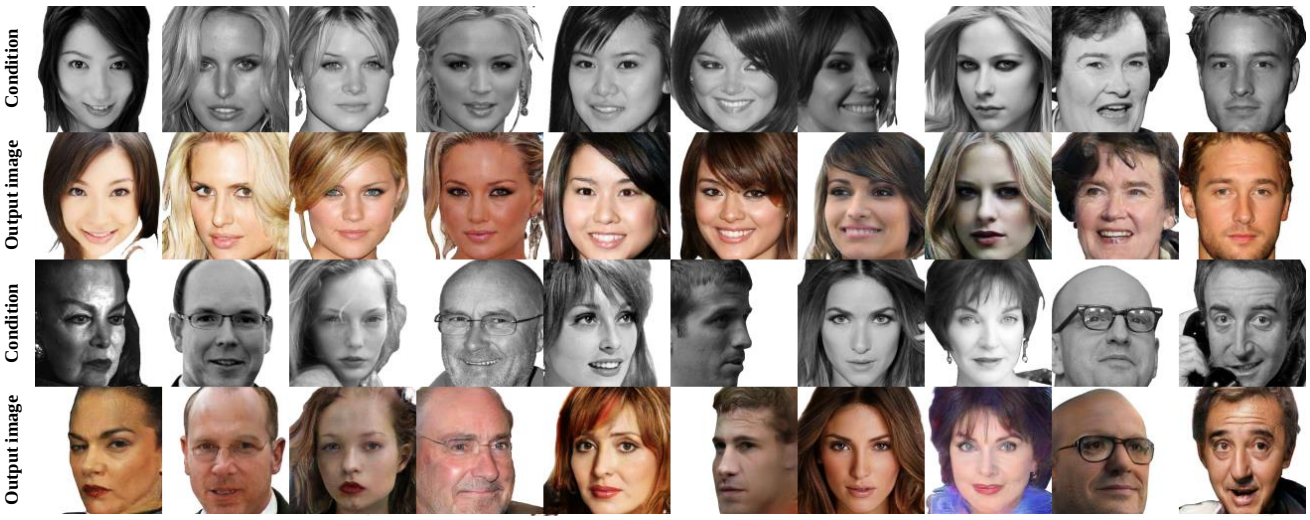
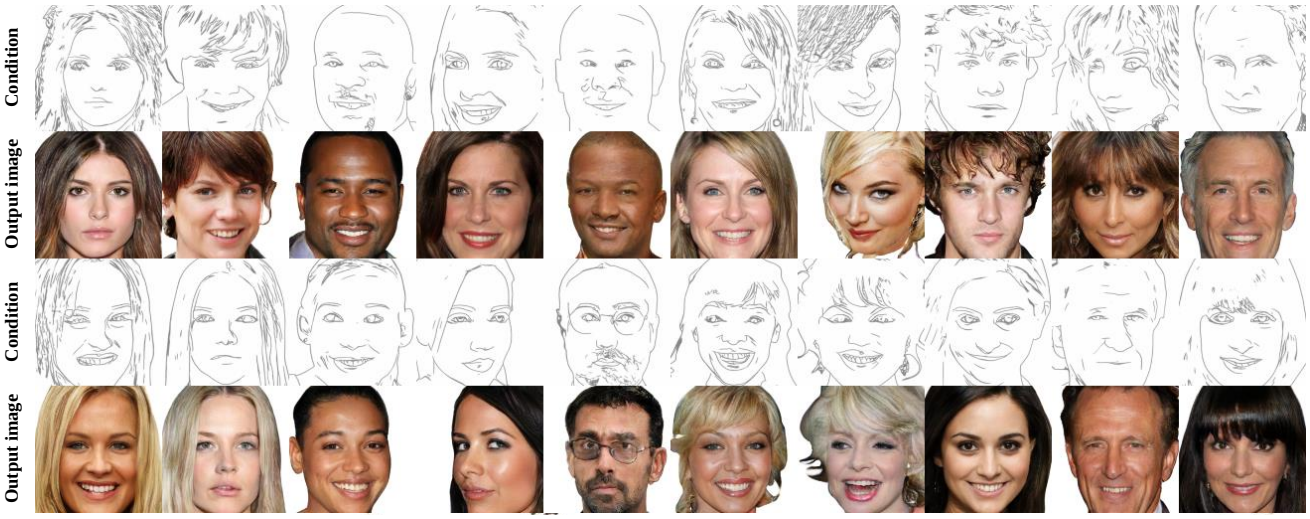


Figure 5: Output images dependent on different condition types on Cats dataset.

(a) CelebA-HQ: Conditioned on color image

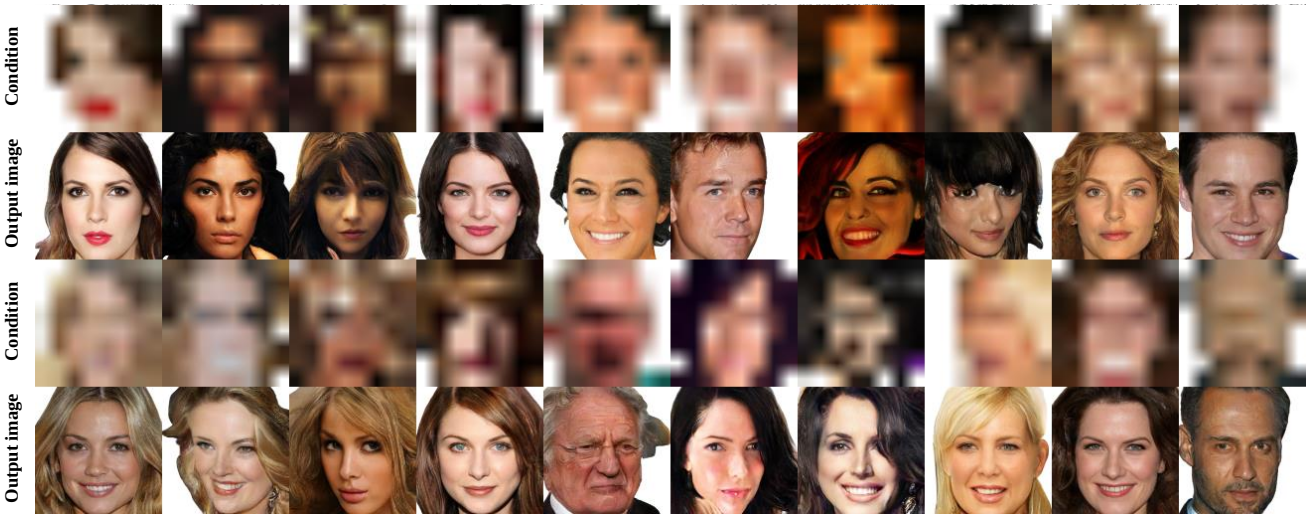


(b) CelebA-HQ: Conditioned on grayscale

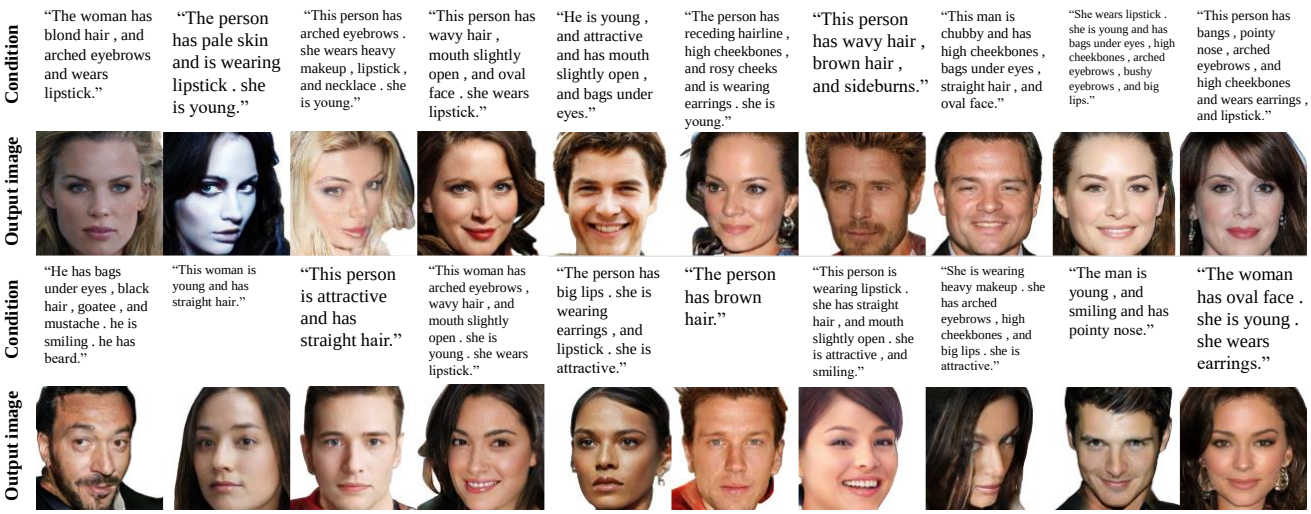


(c) CelebA-HQ: Conditioned on sketch

Figure 6: Uncurated examples from our method trained with various datatset and condition types, each corresponding to 20 random seeds.



(d) CelebA-HQ: Conditioned on low-resolution image



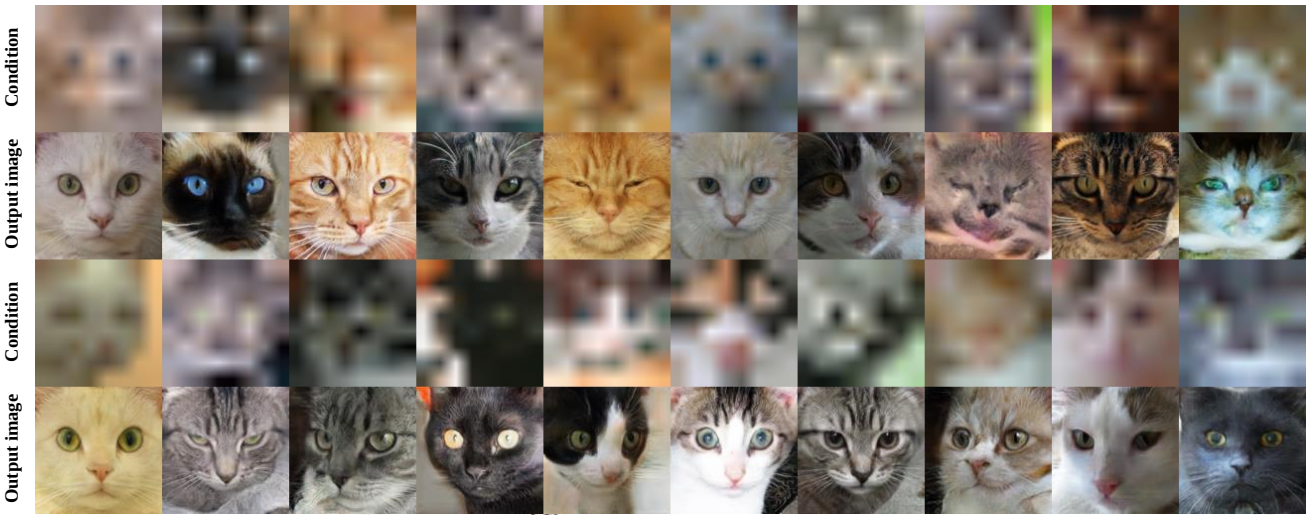
(e) CelebA-HQ: Conditioned on text

(f) CelebA-HQ: Conditioned on grayscale and low-resolution image

Figure 6: Uncurated examples from our method trained with various datatset and condition types, each corresponding to 20 random seeds (continued.).

(g) Cats: Conditioned on color image

(h) Cats: Conditioned on grayscale



(i) Cats: Conditioned on low-resolution image

Figure 6: Uncurated examples from our method trained with various datatset and condition types, each corresponding to 20 random seeds (continued.).

(j) CUB-200: Conditioned on text

(k) FFHQ: Conditioned on color image

Figure 6: Uncurated examples from our method trained with various datatset and condition types, each corresponding to 20 random seeds.