Supplementary Material for "Multimodal Conditional Image Synthesis with Product-of-Experts GANs"

Xun Huang Arun Mallya Ting-Chun Wang Ming-Yu Liu

NVIDIA

In Sec. A of this supplementary material, we describe datasets used in our experiments and how we obtain the annotation of conditional modalities. Sec. B describes implementation details including network architectures, hyperparameters, hardware requirement, and training/inference time. Sec. C presents additional experimental results. Sec. D and Sec. E discusses the potential negative societal impact and the limitation of the proposed approach respectively.

A Datasets

We evaluate the proposed PoE-GAN approach for multimodal conditional image synthesis on three datasets, including MM-CelebA-HQ [24], MS-COCO [11], and a proprietary dataset of landscape images using four modalities including text, segmentation, sketch, and style reference. The style is extracted from ground truth images and the other modalities are obtained from either human annotation or pseudo-labeling methods. We describe details about each dataset in the following.

MM-CelebA-HQ contains 30,000 images of celebrity faces with a resolution of 1024×1024, which are created from the original CelebA dataset [12] using a procedure that improves image quality and resolution by Karras *et al*. [5]. Each image is annotated with text, segmentation, and sketch. While the segmentation maps are annotated by humans [9], the text descriptions are automatically generated from the ground truth attribute labels by Xia *et al*. [24]. On the other hand, sketch maps are generated by Chen *et al*. [4] using the Photoshop edge extractor and sketch simplification [18]. We use the pretrained CLIP [16] text encoder to encode the text before feeding them to the generator. We use the standard train-test split [17,24]. There are 24,000 and 6,000 images in the training set and the test set, respectively. We use images in the original resolution (1024×1024) in our main experiment and use images downsampled to 256×256 in ablation studies.

MS-COCO contains 123,287 images of complex indoor/outdoor scenes, containing various common objects. We use the segmentation maps provided in COCO-Stuff [1] as the ground truth segmentation maps for the images. In MS-COCO, each image has up to 5 text descriptions. We use the pretrained CLIP text encoder to extract a feature vector per description. We additionally annotate each image with a sketch map produced by running HED [25] edge detector followed by a sketch simplification process [18]. We use the 2017 split, which

leads to 118,287 training images and 5,000 test images. We use images resized to 256×256 in our main experiment and to 64×64 in ablation studies.

Landscape is a proprietary dataset containing around 10 million landscape images of resolution higher than 1024×1024. It does not come with any manual annotation and we use DeepLab-v2 [3] to produce pseudo segmentation annotation and HED [25] with sketch simplification [18] to produce pseudo sketch annotation. For the text annotation, we use the CLIP image embedding as the pseudo text embedding to train our model. We randomly choose 50,000 images as the test set and use the rest of the images as the training set. Images are randomly cropped to 1024×1024 during training.

B Implementation details

Our implementation is based on Pytorch [15]. We use the Adam optimizer [8] with $\beta_1 = 0$ and $\beta_2 = 0.99$ and the same constant learning rate for both the generator and the discriminator. The final model weight is given by an exponential moving average of the generator's weights during the course of training. To stabilize GAN training, we employ leaky ReLU [13] with slope 0.2, R_1 gradient penalty [14] with weight 1 and lazy regularization [7] applied every 16 iterations, equalized learning rate [7], anti-aliased resampling [28], and clip the gradient norms at 10. We also limit the range of the Gaussian log variance in product-of-experts layers to $(-\theta, \theta)$ by applying the activation function $\theta \tanh(\frac{\cdot}{\theta})$. We use $\theta = 1$ for the prior expert and $\theta = 10$ for experts predicted from input modalities. We use mixed-precision training in all of our experiments and clamp the output of every convolutional/fully-connected layer to ± 256 [6]. We use a dropout rate of 0.5 when performing modality dropout. The contrastive loss temperature is initialized at 0.3 and learnable during training. We use the `relu5_1` feature in VGG-19 to compute the image contrastive loss.

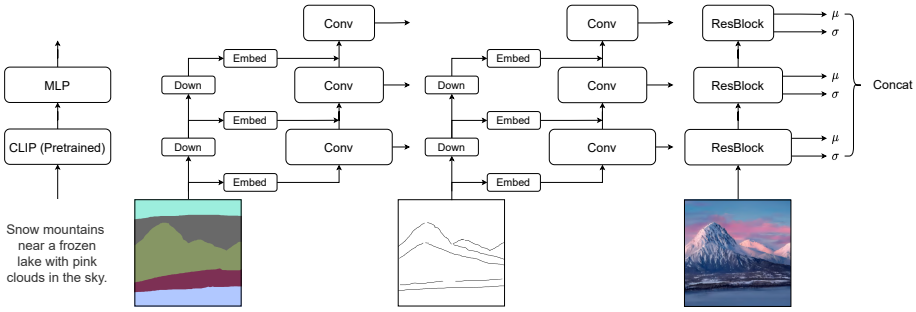
Inspired by NVAE [21], we rebalance the KL terms of different resolutions. The rebalancing weight ω^k in Eq. (9) of the main paper is proportional to the unbalanced KL term and inversely proportional to the resolution:

$$\omega^k \propto \frac{1}{w^k h^k} \mathbb{E}_{p(y_i)} [\mathbb{E}_{p(z^{<k}|y_i)} [D_{\mathrm{KL}}(p(z^k|z^{<k}, y_i) || p(z^k|z^{<k}))]] \,, \tag{1}$$

with the constraint that $\frac{1}{N} \sum_{k=1}^{N} \omega^k = 1$. The rebalancing weights encourage having the amount of information encoded in each latent variable. We use a running average of the rebalancing weights with a decay factor of 0.99.

B.1 Decoder

We introduced our decoder design in Sec. 3.1 of the main paper. Here, we provide additional details. In Global PoE-Net (Fig. 4a of the main paper), each MLP consists of 4 fully-connected layers with a hidden dimension 4 times smaller than the input dimension. Both z^0 and w are 512-dimensional. The output MLP has two layers with a hidden dimension of 512.



(a) Text (b) Segmentation (c) Sketch (d) Style

Fig. 1: Architecture of encoders. We use a pretrained CLIP [16] and an MLP to encoder text, a convolutional network with input skip connections to encode segmentation/sketch maps, and a residual network to encoder style images.

Table 1: Number of parameter of compared models on MM-CelebA-HQ (1024×1024)

StyleGAN2	SPADE-Seg	pSp-Seg	SPADE-Sketch	pSp-Sketch	TediGAN	PoE-GAN
30M	96M	298M	95M	298M	565M	33M

Table 2: Number of parameters of compared models on MS-COCO 2017 (256×256)

StyleGAN2	DF-GAN	DM-GAN+CL	SPADE-Seg	VQGAN	OASIS	SPADE-Sketch	PoE-GAN
25M	12M	32M	280M	798M	94M	95M	142M

In Local PoE-Net (Fig. 4b of the main paper), each CNN similarly contains 4 convolutional layers with the number of filters 4 times smaller than the input channel size. The first and last convolutions have 1×1 kernels, and the convolutions in the middle have a kernel size of 3. The dimension of w is 512. The dimensions of z^k's are described in Sec. B.4.

B.2 Encoders

Fig. 1 shows the architecture of encoders used in our generator to encode each modality. The text encoder (Fig. 1a) is a 4-layer MLP with dimension 512 that processes the CLIP embedding of a caption. The segmentation and sketch encoders are CNNs with skip connections from the input, which are illustrated in Fig. 1b and Fig. 1c. The segmentation or sketch map is downsampled multiple times. The embeddings of the downsampled map are added to the intermediate outputs of the corresponding convolutional layers. The intermediate outputs of convolutional layers are provided to the decoder via skip connections. The style encoder for encoding the style image is a residual network with instance normalization [20]. As shown in Fig. 1d, we obtain the style code by concatenating the mean and standard deviation of the output of every residual block.

Table 3: Hyper-parameters on different datasets

Hyper-parameters	MM-CelebAHQ (1K × 1K)	MM-CelebAHQ (256 × 256)	MS-COCO (256 × 256)	MS-COCO (64 × 64)	Landscape (1K × 1K)
Learning rate	0.003	0.003	0.004	0.004	0.004
Batch size	64	64	256	128	768
Text KL weight	0.01	0.1	0.01	0.1	0.05
Segmentation KL weight	0.01	0.1	0.01	0.1	0.1
Sketch KL weight	0.1	1	0.1	1	1
Style KL weight	0.0005	0.005	0.001	0.01	0.01
Image contrastive loss weight	0.3	0.3	3	3	3
Text contrastive loss weight	0.3	0.3	0.3	0.3	0.3
Base # channels for Dec/Dis	16	64	128	256	32
Maximum # channels for Dec/Dis	512	512	1024	512	1024
Base # channels for Latent	2	16	16	64	2
Maximum # channels for Latent	32	64	64	64	32

Table 4: Hardware and training/inference speed on different datasets. We train all models using NVIDIA Tesla V100 GPUs, except for the Landscape model which is trained using NVIDIA Ampere A100 GPUs with 80 GB of memory. The inference time is evaluated on a workstation with a single NVIDIA TITAN RTX GPU

	MM-CelebAHQ (1K × 1K)	MM-CelebAHQ (256 × 256)	MS-COCO (256 × 256)	MS-COCO (64 × 64)	Landscape (1K × 1K)
Number of GPUs	16	8	32	8	256
Training time	71h	35h	85h	76h	101h
Inference time (per image)	0.07s	0.04s	0.06s	0.02s	0.12s

B.3 Discriminator

As shown in Fig. 5c of the main paper, our discriminator encodes the image and the other modalities into multiscale feature maps and computes the MPD loss at each scale. We use a residual network to encode the image. For encoding other modalities, we use the same architecture as described in the previous section (Sec. B.2 and Fig. 1). However, the parameters are not shared with those encoders used in the generator. For encoders (the style encoder or the text encoder) that only outputs a single feature vector, we spatially replicate the feature vector to different resolutions.

B.4 Hyper-parameters

Tab. 3 shows the hyper-parameters used on all the benchmark datasets, including learning rate, batch size, loss weights, and channel size. The decoder and

Table 5: Comparison of text-to-image synthesis on MM-CelebA-HQ (256×256). ↑: the higher the better, ↓: the lower the better

Method	AttnGAN [26]	ControlGAN [10]	DF-GAN [19]	DM-GAN [30]	TediGAN [24]	M6-UFC [29]	Ours
FID ↓	125.98	116.32	137.60	131.05	106.37	66.72	**13.71**
LPIPS ↑	0.512	0.522	0.581	0.544	0.456	0.448	**0.583**

discriminator have the same channel size, which is always 4 times larger than the channel size of the modality encoders. We hence omit the channel size of modality encoders from the table. The layer operating at the highest resolution always has the smallest number of channels (denoted as "base # channels for DecDis"). The number of channels doubles while the resolution halves until it reaches a maximum number (denoted as "maximum # channels for DecDis"). The "Base # of channels for Latent" refers to the channel size of the feature map z_i^N to the decoder extracted from the spatial modalities (segmentation and sketch) in the highest resolution (the largest value of k is N). We also note that the channel number of μ_i^k or σ_i^k equals that of z_i^k. As explained in Sec. B.2, we double the channel size as we halve the spatial resolution. The channel number for z_i^k doubles until it reaches the "Maximum # of channels for Latent," the channel number for z_i^0.

B.5 Hardware and speed

We report the computation infrastructure used in our experiments Tab. 4. We also report the training and inference speed of our model for different datasets in the table.

C Additional results

We provide additional experimental results in this section, including more comparison with baselines and more analysis on the proposed approach. Please check the accompanying video for additional results on the landscape dataset.

C.1 Model size comparison

In Tab. 1 and Tab. 2, we compare the number of parameters used in PoE-GAN and in our baselines. We show that PoE-GAN does not use significantly more parameters — actually it uses fewer parameters than some of the single-modal baselines, although PoE-GAN is trained for a much more challenging task. This shows that our improvement does not come from using a larger model.

C.2 Comparison on MM-CelebA-HQ (256×256)

In Tab. 5, We compare PoE-GAN trained on the 256×256 MM-CelebA-HQ dataset with the text-to-image baselines reported in TediGAN [24] and M6-UFC [29]. Our model achieves a significantly lower FID than previous methods.

C.3 Additional comparison with TediGAN

We provide a more detailed comparison between PoE-GAN and TediGAN, the only existing multimodal image synthesis method that can produce high-resolution images. TediGAN mainly contains four steps: 1) encodes different modalities into the latent space of StyleGAN, 2) mixes the latent code according to a hand-designed rule, 3) generates the image from the latent code using StyleGAN, and 4) iteratively refines the image. It has several disadvantages compared with our PoE-GAN:

1. TediGAN relies on a pretrained unconditional generator, and its image quality is upper bounded by that unconditional model. This is not ideal because unconditional models usually produce images of lower quality than conditional ones. On the other hand, PoE-GAN learns conditional and unconditional generation simultaneously, and its FID improves when more conditions are provided, as shown in Tab. 1 of the main paper.
2. TediGAN uses a handcrafted rule to combine the latent code from different modalities. Specifically, the StyleGAN latent space contains 14 layers of latent vectors. TediGAN uses the top-layer latent vectors from one modality and the bottom-layer latent vectors from another modality when combining two modalities. This combination rule cannot be generalized to other generator architectures and other modalities. In contrast, we use product-of-experts with learned parameters to combine different modalities which is more general.
3. Sampling from TediGAN is very slow due to its instance-level optimization. It takes 51.2 seconds to generate a 1024×1024 image with TediGAN, while PoE-GAN only needs 0.07 seconds.

C.4 Training PoE-GAN with a single modality

Although PoE-GAN is designed for the multimodal conditional image synthesis task, it can be trained in a single modality setting. That is, we can train a pure segmentation-to-image model, a pure sketch-to-image model, or a pure text-to-image model using the same PoE-GAN model. Here, we compare a PoE-GAN model trained using multiple modalities with the same PoE-GAN model but trained using one single modality. We compare their performance when applied to convert the user input in a single modality to the output image. This experiment helps understand the penalty the model pays for the multimodal synthesis capability.

As shown in Tab. 6 and Tab. 7, the model trained for a specific modality always slightly outperforms the joint model when conditioned on that modality. This indicates that the increased task complexity of an additional modality outweighs the benefits of additional annotations from that modality. This result also shows that the improvement of PoE-GAN over state-of-the-art unimodal image synthesis methods comes from our architecture and training scheme rather than additional annotations.

Table 6: Comparison on MM-CelebA-HQ (256×256) using FID. We compare our PoE-GAN trained using all modalities with PoE-GAN trained using a single modality. Note that PoE-GAN trained using a single modality can also be used for unconditional synthesis and we also report the achieved FID in the table

	Uncond	Text	Seg	Sketch	All
Ours (Uncond)	**13.0**	—	—	—	—
Ours (Text)	13.8	**13.4**	—	—	—
Ours (Seg)	14.2	—	**10.9**	—	—
Ours (Sketch)	14.2	—	—	**9.5**	—
Ours (All)	14.9	13.7	12.9	9.9	**8.5**

Table 7: Comparison on MS-COCO 2017 (64×64) using FID. We compare our PoE-GAN trained using all modalities with PoE-GAN trained using a single modality. Note that PoE-GAN trained using a single modality can also be used for unconditional synthesis and we also report the achieved FID in the table

	Uncond	Text	Seg	Sketch	All
Ours (Uncond)	**23.3**	—	—	—	—
Ours (Text)	24.0	**21.1**	—	—	—
Ours (Seg)	25.5	—	**16.3**	—	—
Ours (Sketch)	24.7	—	—	**24.7**	—
Ours (All)	26.6	22.2	17.1	30.2	**17.1**

C.5 Additional qualitative examples

In Figs. 3 to 7, we show that PoE-GAN can generate diverse images when conditioned on two different input modalities. We show additional qualitative comparison of text-to-image synthesis and segmentation-to-image synthesis on MS-COCO in Figs. 8 and 9 respectively. Figs. 10 to 12 show uncurated samples generated unconditionally on MM-CelebA-HQ, MS-COCO, and Landscape.

Huge ocean waves clash into rocks.

A beach with black sand and palm trees.

Fig. 2: Examples generated by PoE-GAN when conditioned on contradictory segmentation and text inputs. The text input is simply ignored.

D Potential negative societal impacts

Image synthesis networks can help people express themselves and artists create digital content, but they can undeniably also be misused for visual misinformation [23]. Enabling users to synthesize images using multiple modalities makes it even easier to create a desired fake image. We encourage research that helps detect or prevent these potential negative misuses. We will provide implementations and training data to help forensics researchers detect fake images [22,2] and develop effective schemes for watermarking networks and training data [27]. While image synthesis methods can help people express themselves and artists create digital content for movies and games, undeniably, they can be maliciously used for visual misinformation [23]. Our method allows users to synthesize images using multiple modalities, which makes it even easier to create the ideal fake image. We are aware of its potential negative impacts and are taking preventive steps, including providing reference implementations for deep fake detection research [22] and developing effective watermarking schemes [27].

E Limitation

We find that the PoE-GAN model does not work well when conditioned on contradictory multimodality inputs. For example, when the segmentation and text are contradictory to each other, the text input is usually ignored, as shown in Fig. 2. In the product-of-experts formulation, an expert with a larger variance will have a smaller influence on the product distribution, and we indeed find the variance of the text expert is usually larger than that of the segmentation expert, which explains the behavior of our model.

In addition, as discussed in Sec. C.4, the PoE-GAN model still pays the penalty in terms of a higher FID score as achieving the multimodal conditional image synthesis capability. This indicates room for improvement in the fusing of multiple modalities, and we leave this for future work

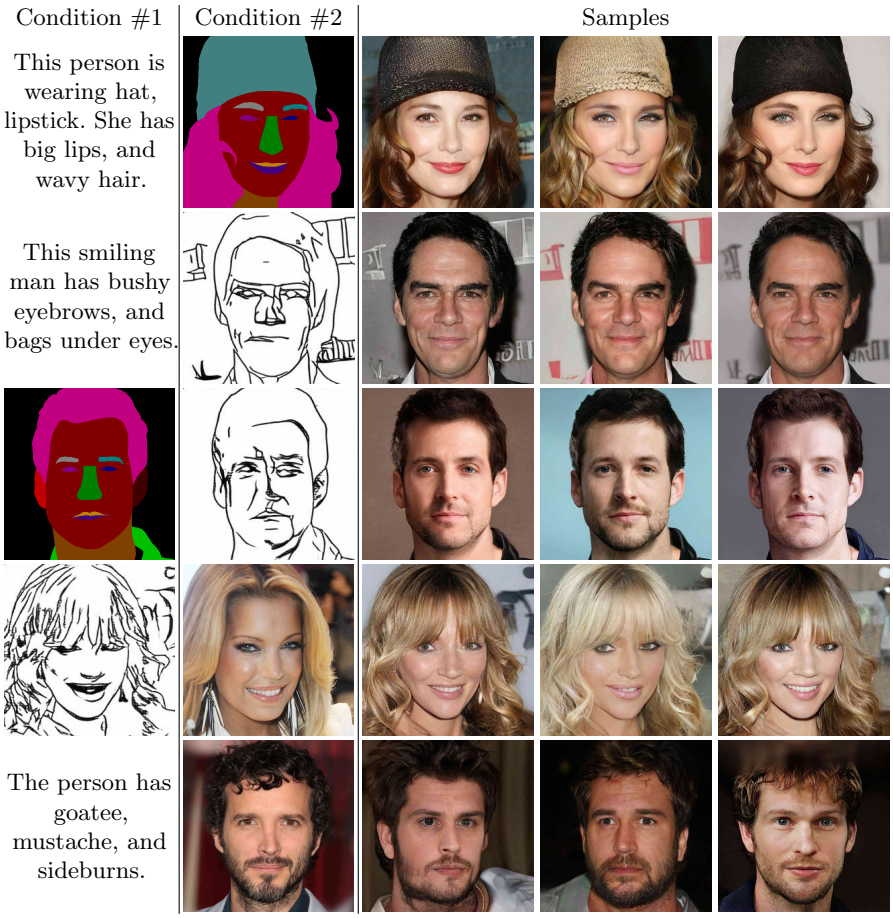


Fig. 3: Examples of multimodal conditional image synthesis on MM-CelebA-HQ. We show three random samples from PoE-GAN conditioned on two modalities.

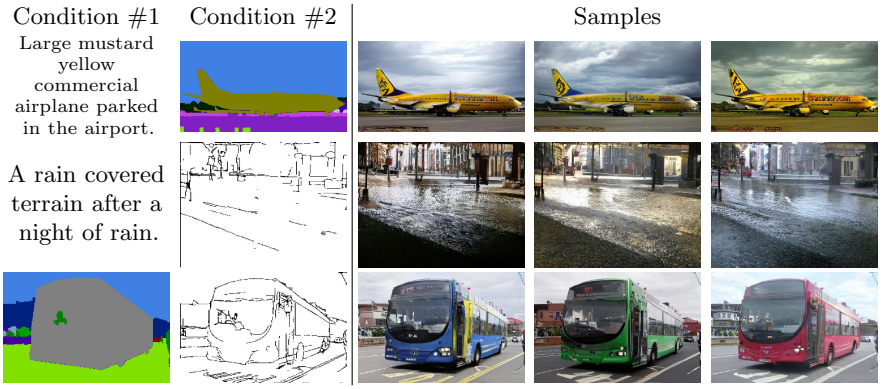


Fig. 4: Examples of multimodal conditional image synthesis on MS-COCO. We show three random samples from PoE-GAN conditioned on two modalities (from top to bottom: text + segmentation, text + sketch, and segmentation + sketch).

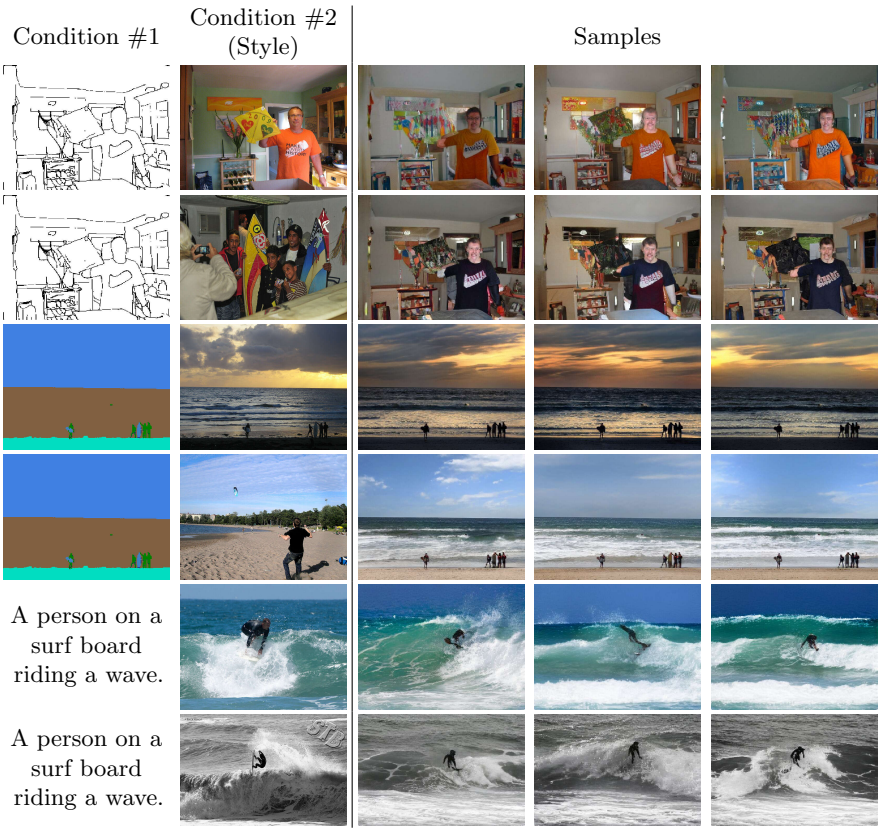


Fig. 5: Examples of multimodal conditional image synthesis on MS-COCO. We show three random samples from PoE-GAN conditioned on two modalities, one being text/segmentation/sketch and another being style reference.

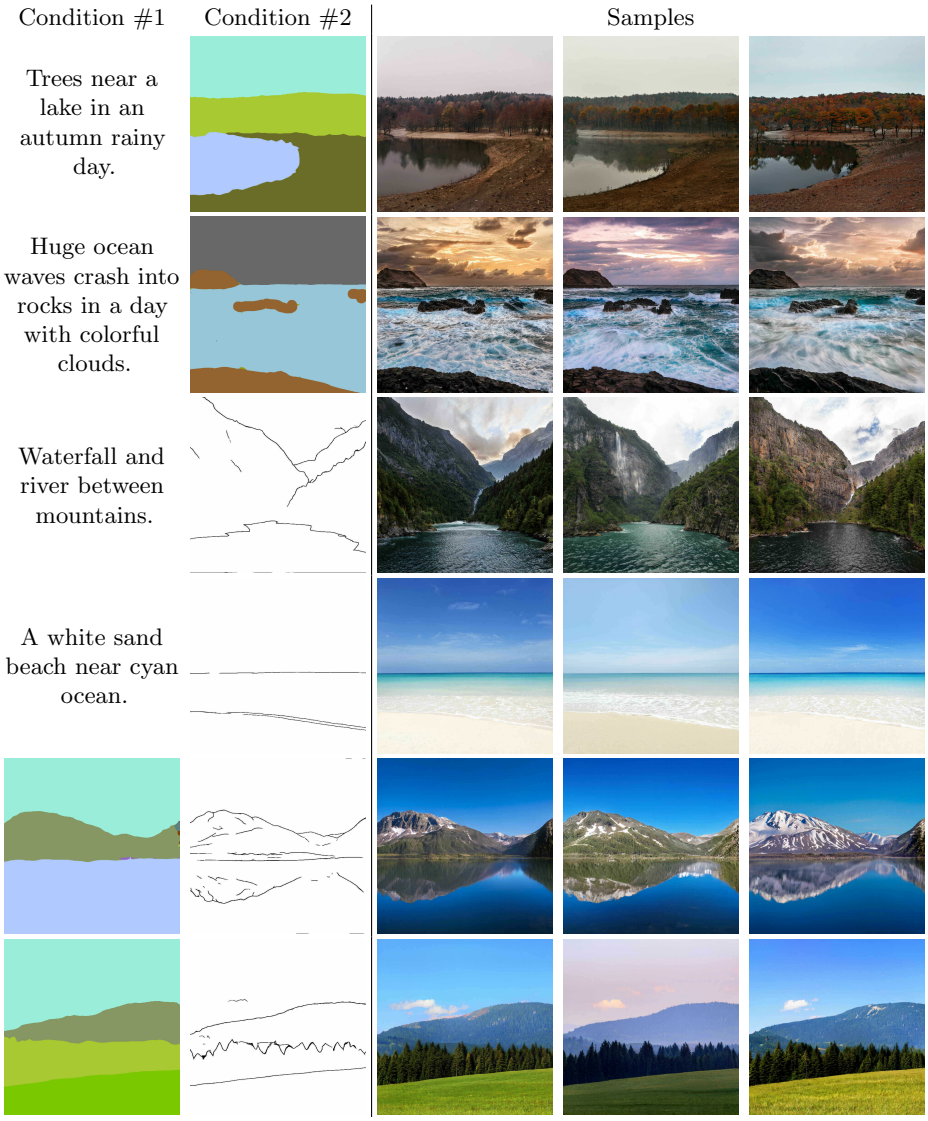


Fig. 6: Examples of multimodal conditional image synthesis on Landscape. We show three random samples from PoE-GAN conditioned on two modalities (from top to bottom: text + segmentation, text + sketch, and segmentation + sketch).

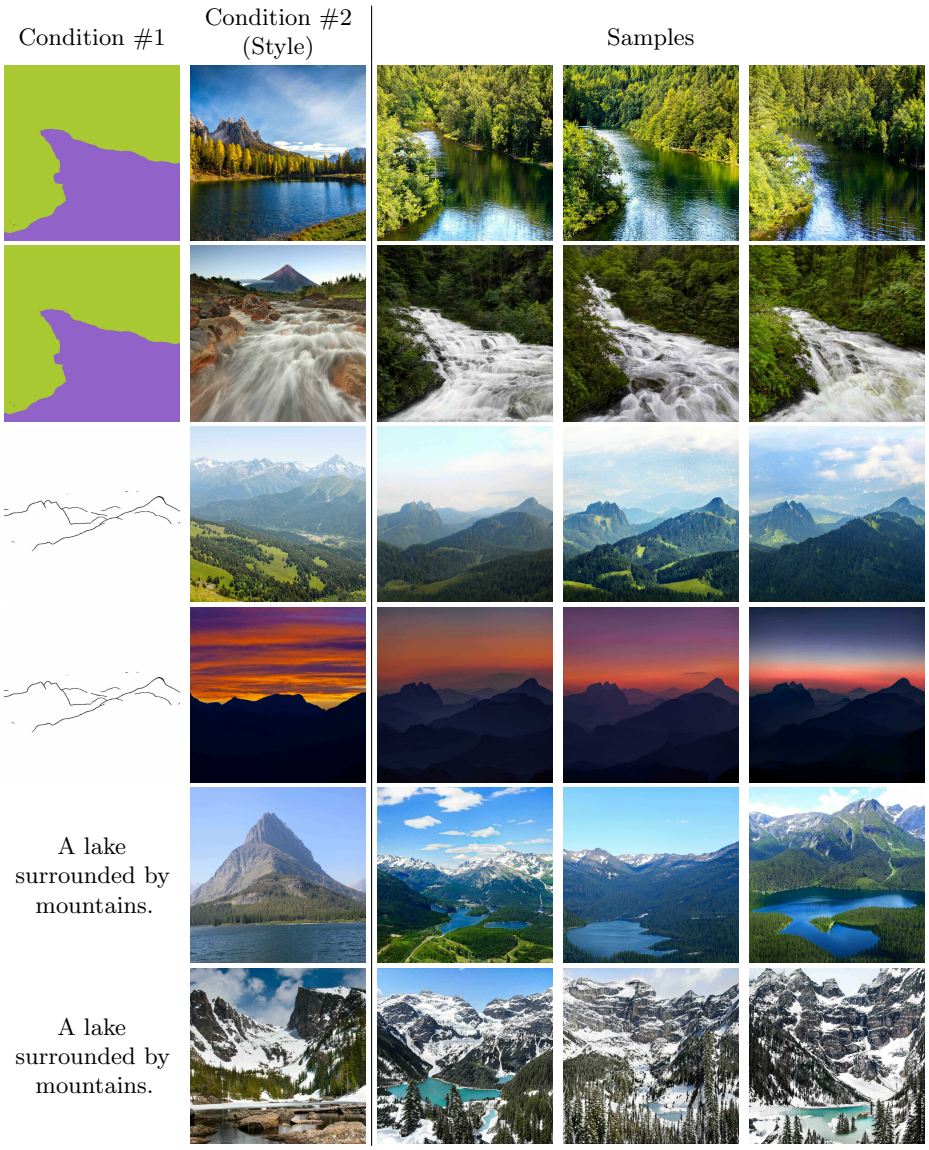


Fig. 7: Examples of multimodal conditional image synthesis on Landscape. We show three random samples from PoE-GAN conditioned on two modalities, one being text/segmentation/sketch and another being style reference.

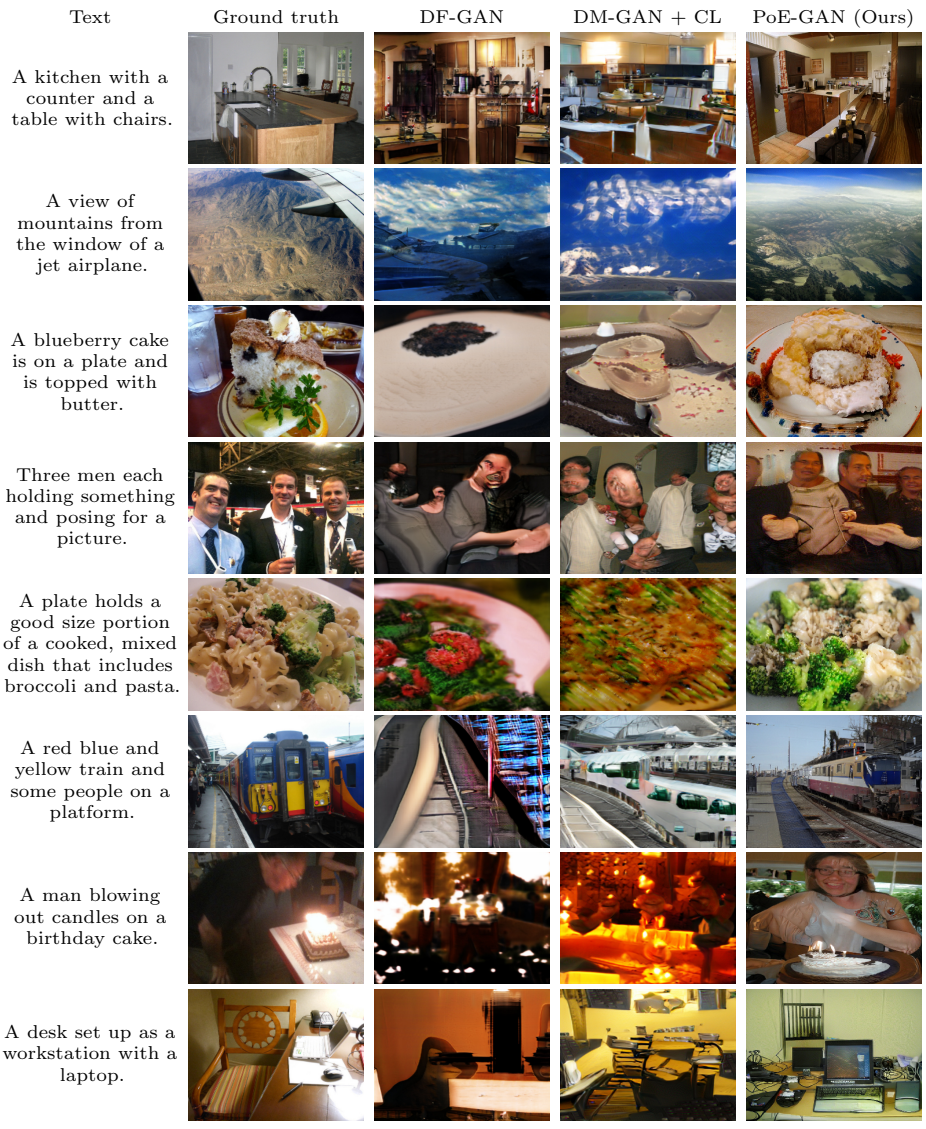


Fig. 8: Additional visual comparison of text-to-image synthesis on MS-COCO 2017.

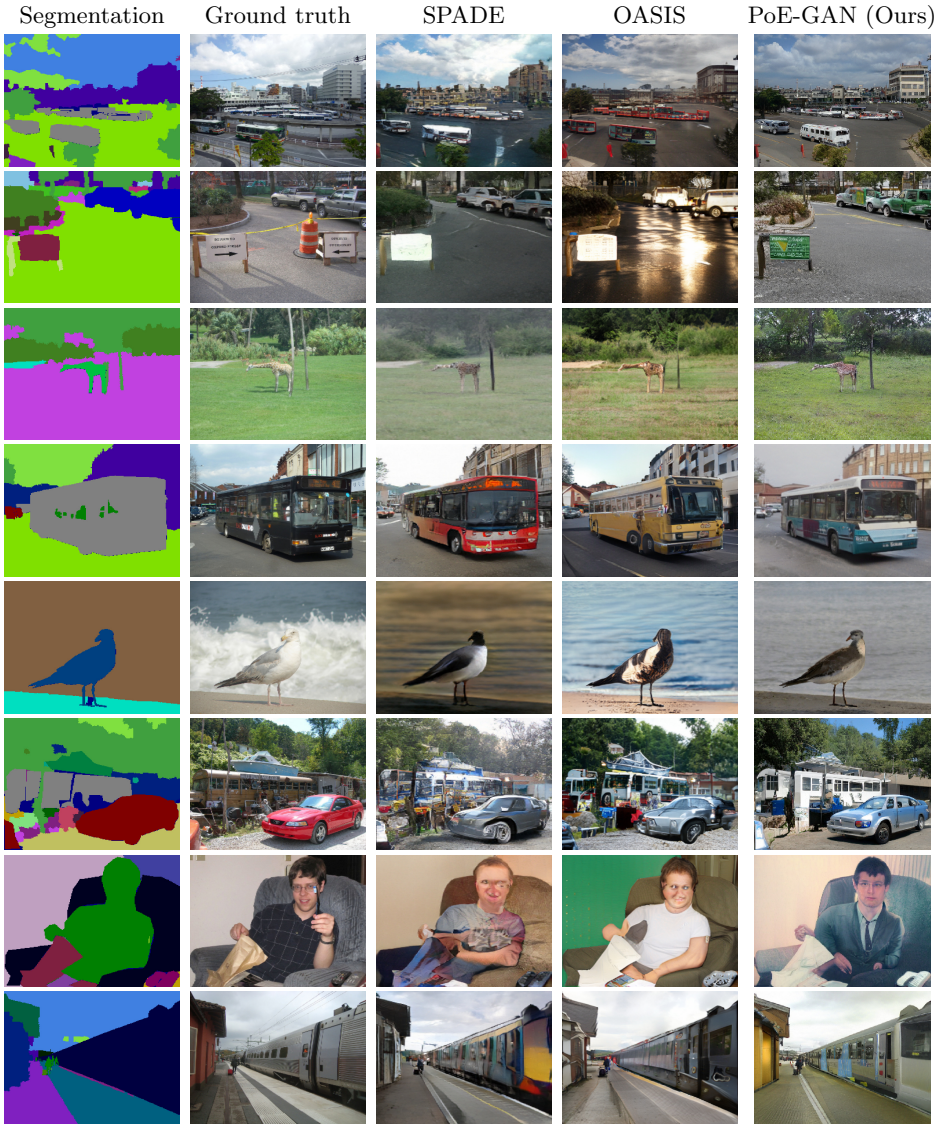


Fig. 9: Additional visual comparison of segmentation-to-image synthesis on MS-COCO 2017.

Fig. 10: Uncurated unconditional results on the 1024×1024 MM-CelebA-HQ dataset.

Fig. 11: Uncurated unconditional results on the 256×256 MS-COCO dataset.

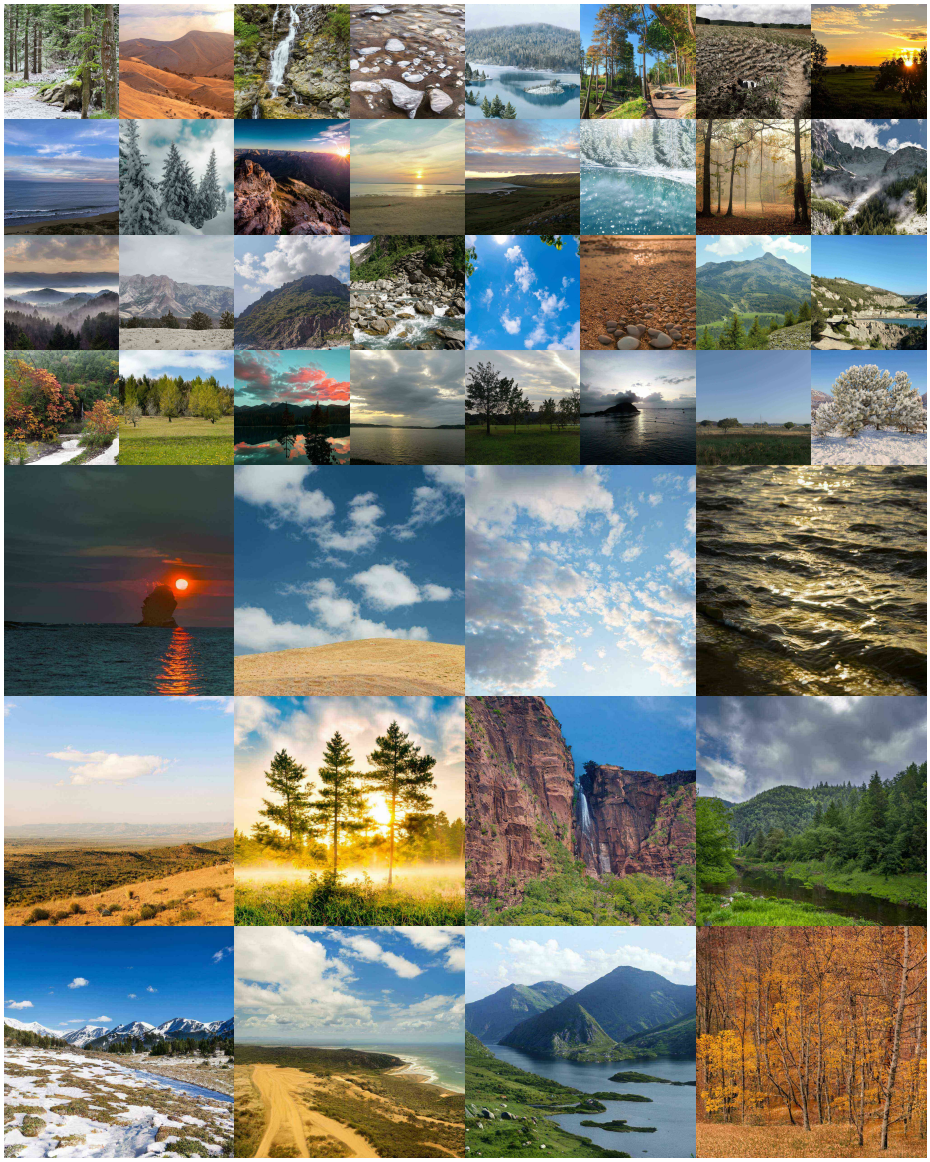


Fig. 12: Uncurated unconditional results on the 1024×1024 landscape dataset.

References

1. Caesar, H., Uijlings, J., Ferrari, V.: COCO-Stuff: Thing and stuff classes in context. In: CVPR (2018) 1
2. Chai, L., Bau, D., Lim, S.N., Isola, P.: What makes fake images detectable? understanding properties that generalize. In: ECCV (2020) 8
3. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence (2017) 2
4. Chen, S.Y., Su, W., Gao, L., Xia, S., Fu, H.: DeepFaceDrawing: Deep generation of face images from sketches. ACM Transactions on Graphics (TOG) (2020) 1
5. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. In: ICLR (2018) 1
6. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training generative adversarial networks with limited data. In: NeurIPS (2020) 2
7. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: CVPR (2020) 2
8. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2014) 2
9. Lee, C.H., Liu, Z., Wu, L., Luo, P.: Maskgan: Towards diverse and interactive facial image manipulation. In: CVPR (2020) 1
10. Li, B., Qi, X., Lukasiewicz, T., Torr, P.H.: Controllable text-to-image generation. In: NeurIPS (2019) 5
11. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: ECCV (2014) 1
12. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: ICCV (2015) 1
13. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: ICML (2013) 2
14. Mescheder, L., Geiger, A., Nowozin, S.: Which training methods for GANs do actually converge? In: ICML (2018) 2
15. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: PyTorch: An imperative style, high-performance deep learning library. In: NeurIPS (2019) 2
16. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021) 1, 3
17. Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., Cohen-Or, D.: Encoding in style: a StyleGAN encoder for image-to-image translation. In: CVPR (2021) 1
18. Simo-Serra, E., Iizuka, S., Sasaki, K., Ishikawa, H.: Learning to simplify: fully convolutional networks for rough sketch cleanup. ACM Transactions on Graphics (TOG) (2016) 1, 2
19. Tao, M., Tang, H., Wu, S., Sebe, N., Jing, X.Y., Wu, F., Bao, B.: DF-GAN: Deep fusion generative adversarial networks for text-to-image synthesis. arXiv preprint arXiv:2008.05865 (2020) 5
20. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In: CVPR (2017) 3

21. Vahdat, A., Kautz, J.: NVAE: A deep hierarchical variational autoencoder. In: NeurIPS (2020) 2
22. Wang, S.Y., Wang, O., Zhang, R., Owens, A., Efros, A.A.: CNN-generated images are surprisingly easy to spot... for now. In: CVPR (2020) 8
23. Westerlund, M.: The emergence of deepfake technology: A review. Technology Innovation Management Review (2019) 8
24. Xia, W., Yang, Y., Xue, J.H., Wu, B.: TediGAN: Text-guided diverse face image generation and manipulation. In: CVPR (2021) 1, 5
25. Xie, S., Tu, Z.: Holistically-nested edge detection. In: ICCV (2015) 1, 2
26. Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., He, X.: AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In: CVPR (2018) 5
27. Yu, N., Skripniuk, V., Abdelnabi, S., Fritz, M.: Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In: ICCV (2021) 8
28. Zhang, R.: Making convolutional networks shift-invariant again. In: ICML (2019) 2
29. Zhang, Z., Ma, J., Zhou, C., Men, R., Li, Z., Ding, M., Tang, J., Zhou, J., Yang, H.: M6-UFC: Unifying multi-modal controls for conditional image synthesis. In: NeurIPS (2021) 5
30. Zhu, M., Pan, P., Chen, W., Yang, Y.: DM-GAN: Dynamic memory generative adversarial networks for text-to-image synthesis. In: CVPR (2019) 5