Image Generators with Conditionally-Independent Pixel Synthesis

Supplementary Materials

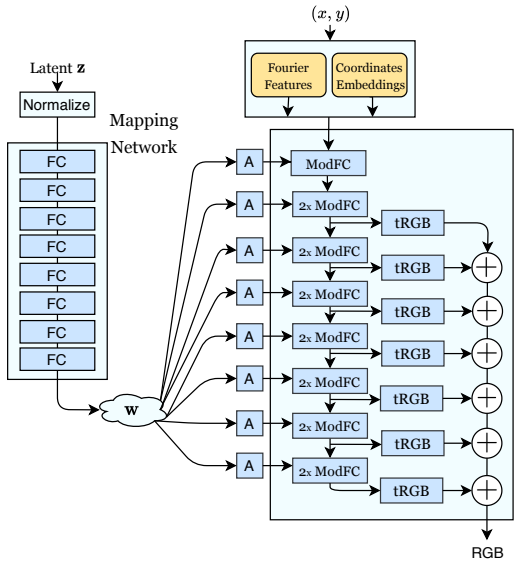


Figure 1: The diagram of the CIPS generator (default version).

Modification	# parameters (mln)
CIPS-base	43.8
CIPS-NE	10.2
CIPS-default	45.9
StyleGANv2	30.0

Table 1: The number of parameters for different version of the CIPS generator. For reference, the number of parameters within the StyleGANv2 generator is also given.

1. Architecture details

In this section we provide additional information about the default version of our CIPS generator (Fig. 1). In total, its backbone contains 15 fully connected layers. The first layer projects concatenated coordinate embeddings and Fourier features into the joint space with the dimension of 512. Next, the following layer pattern is repeated seven times: the representation is put through two modulated fully-connected layers and a projection to RGB color space is computed. The projections coming from the seven iterations are summed together to create the final image. Similarly to StyleGANv2 we add properly broadcasted noise maps of size $H \times W$ in every ModFC layer (not shown in Fig. 1). We also adopt from StyleGANv2 other settings, including leaky ReLU activation with the slope 0.2, NTK-parameterization, exponential moving average for weights, antialiased bilinear down/upsampling.

Our model is trained with a standard non-saturating logistic GAN loss with R_1 penalty [6] applied to the discriminator D. The discriminator has a residual architecture, described in [3] (we have deliberately kept the discriminator architecture intact). Networks were trained by Adam optimizer [4] with learning rate 2×10^{-3} and hyperparameters: $\beta_0 = 0$, $\beta_1 = 0.99$, $\epsilon = 10^{-8}$.

The number of parameters for the different modifications of the CIPS generator discussed in the paper are given in Tab. 1.

2. Patch-based generation

To show one benefit of coordinate-based approach, we demonstrate the results of *memory-constrained* training, where the discriminator observes patches at lower resolution than the full image (inspired by the GRAF system [8]). Since pixel generation is conditionally-independent, at each iteration only low-resolution patches need to be generated. Thus, only the following $K \times K$ patch is synthesized and submitted to the discriminator:

$$P_{K,\sigma}(u, v) = \{G(u + i\sigma, v + j\sigma; \mathbf{z}) \mid (i, j) \in \texttt{mgrid}(K, K)\},$$

where $0 \leq u < W - (K-1)\sigma$ and $0 \leq v < H - (K-1)\sigma$ are the coordinates of the corner pixel of the patch. For $\sigma = 1$ this produces dense patch, while for $\sigma > 1$ a dilated patch with increased receptive field is obtained. Applying this patch sampling to real images before putting them into the discriminator may be thought of as an example of a differentiable augmentation, the usefulness of which was recently proved by [1, 10].

Dataset	Image size	Patch size	FID
FFHQ	256	64	11.79
		128	9.08
		256	4.38
	1024	256	11.57
		512	10.07
LSUN-Churches	256	64	11.53
		128	7.08
		256	2.92

Table 2: Frechet Inception Distance (FID) values for CIPS models trained on patches of varying receptive field and fixed resolution. The results for patch-based training are worse than the default training procedure, in which the discriminator observes the full image.

Tab. 2 reports the quality (FID) for CIPS generators trained on patches of sizes 64×64 and 128×128, while the resolution of full images equals 256×256. Fig. 2 shows the outputs of models, trained with the patch-based pipeline. In our experiments, training with smaller size of patches degrades the overall quality of resulting samples.

For images with high resolution 1024×1024, CIPS generator was trained progressively starting from 256×256 initialization. See Tab. 2 and Fig. 6 for results.

3. Additional results

3.1. Panorama synthesis

As CIPS is built upon a coordinate grid, it can relatively easily use non-Cartesian grids. To show this, we thus adopt a cylindrical system to produce landscape panoramas. The training setup is as follows. We uniformly sample a 256×256 crop from the cylindrical coordinate grid and train the generator to produce images using these coordinate crops as inputs. A similar idea was also explored in [5]. We note, however, that during training we do not use any real panoramas in contrast to other coordinate-based COCO-GAN model [5]. Fig. 3a and 3b provide examples of panorama samples obtained with the resulting model.

As each pixel is generated from its coordinates and style vector only, our architecture admits pixel-wise style interpolation (Fig. 3c). In these examples, the style vector blends between the central part (the style of Fig. 3a) and the outer part (the style of 3b). We also demonstrate more samples of cylindrical panoramas in Fig. 8.

3.2. Samples from CIPS

In Fig. 7 and 6, we provide additional samples from CIPS generators trained on different datasets.

Although we do not apply mixing regularization [2] at train time, our model is still capable of layer-wise combination of latent variables at various depth (see Fig. 10). The examples suggest that similarly to StyleGAN, different layers of CIPS control different aspects of images.

LSUN-Churches

FFHQ

Figure 2: Samples from CIPS generators learned with memory-constrained patch-based training. Within every grid, the top row contains images from models trained with patches of size 128×128 and the bottom row represents outputs from training on 64×64 patches. While the samples obtained with such memory-constrained training are meaningful, their quality and diversity are worse compared to standard training.

Generator	$\mathcal{W}$	$\mathcal{W}+$
StyleGANv2	0.63	0.75
CIPS	**0.70**	**0.81**

Table 3: SSIM for random images from CelebA-HQ projected into the latent space of two generators. CIPS obtains a better result both in case of encoding to a single style vector and when projecting to an extended style space.

3.3. Projection into the latent space

We compared the optimization-based inversion in CIPS and in StyleGANv2 trained on FFHQ-256 for 35 random images from CelebA-HQ. During inversion we minimize over the L_1+VGG loss over the latent space. The results in terms of structural similarity (SSIM) are reported in Tab. 3. The qualitative comparison showing a typical advantage of

(a)

(b)

(c)

Figure 3: Panorama blending. We linearly blend two upper images from CIPS generator trained on the Landscapes dataset with a cylindrical coordinate system. The resulting image contains elements from both original panoramas: land and water integrated naturally.

Figure 4: Results of encoding an image to the latent space. Left: input image; middle: CIPS $\mathcal{W}+$ inversion; right: StyleGANv2 $\mathcal{W}+$ inversion. Our model preserves more fine-grained details (e.g. note the earring).

our method is illustrated in Fig. 4. We hypothesize CIPS gets better scores due to both the greater number of layers and pixel-wise computation prior.

4. Nearest neighbors

To assess the generalization ability of CIPS architecture, we also show the samples from the model trained on the FFHQ face dataset alongside the most similar faces from the train dataset. To mine the most similar faces, we extract faces using the MTCNN model [9], and then compute their embeddings using FaceNet [7] (the public implementation of these models[1] was used). Fig. 9 shows five nearest neighbors (w.r.t. FaceNet descriptors) for each samples. The samples generated by the model are clearly not duplicates of the training images.

[1] https://github.com/timesler/facenet-pytorch

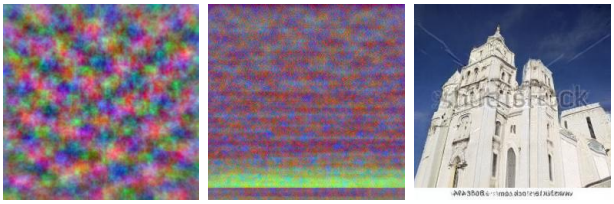


Figure 5: Visualisation of three main principal components of coordinate embeddings for CIPS models, trained on Landscapes (left) and LSUN-Churches (center). As these datasets are not as aligned as the face dataset, there is less recognizable structure in the learned coordinate embeddings. The bottom horizontal structure in the LSUN-Churches case is likely due to frequent watermark pattern in the dataset (a sample from the model with such watermark is shown on the right).

5. Coordinate embeddings

We also run the Principle Components Analysis (PCA) for coordinate embeddings of models trained on Landscapes and LSUN-Churches images (similar pattern for the FFHQ dataset is shown in the main paper). Fig. 5 provides the visualisation for the three main components. Note, that as these datasets are as aligned as FFHQ, there is considerably less spatial structural information in the learned embeddings.

References

[1] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila. Training Generative Adversarial Networks with Limited Data. In *Proc. NeurIPS*. Curran Associates, Inc.,

2020. 1

[2] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proc. CVPR*, pages 4396–4405, 2019. 2

[3] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan. In *Proc. CVPR*, pages 8107–8116, 2020. 1

[4] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations, ICLR*, 2015. 1

[5] C. H. Lin, C. Chang, Y. Chen, D. Juan, W. Wei, and H. Chen. Coco-gan: Generation by parts via conditional coordinating. In *Proc. ICCV*, pages 4511–4520, 2019. 2

[6] L. Mescheder, A. Geiger, and S. Nowozin. Which Training Methods for GANs do actually Converge? In J. Dy and A. Krause, editors, *Proc. ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 3481–3490, Stockholmsmässan, Stockholm Sweden, July 2018. PMLR. 1

[7] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015. 3, 7

[8] K. Schwarz, Y. Liao, M. Niemeyer, and A. Geiger. GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. In *Proc. NeurIPS*. Curran Associates, Inc., 2020. 1

[9] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 3

[10] S. Zhao, Z. Liu, J. Lin, J.-Y. Zhu, and S. Han. Differentiable Augmentation for Data-Efficient GAN Training. In *Proc. NeurIPS*. Curran Associates, Inc., 2020. 1

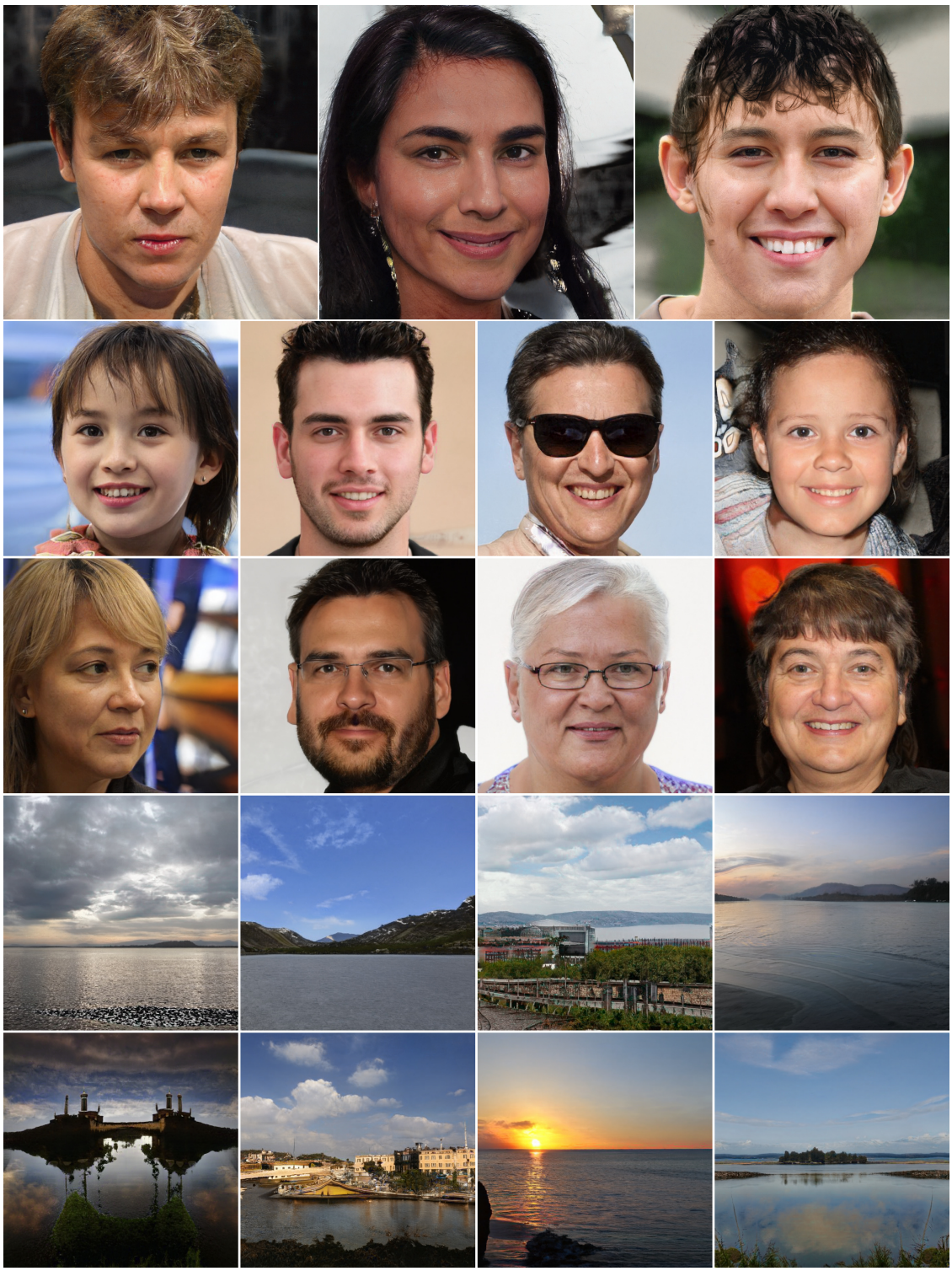
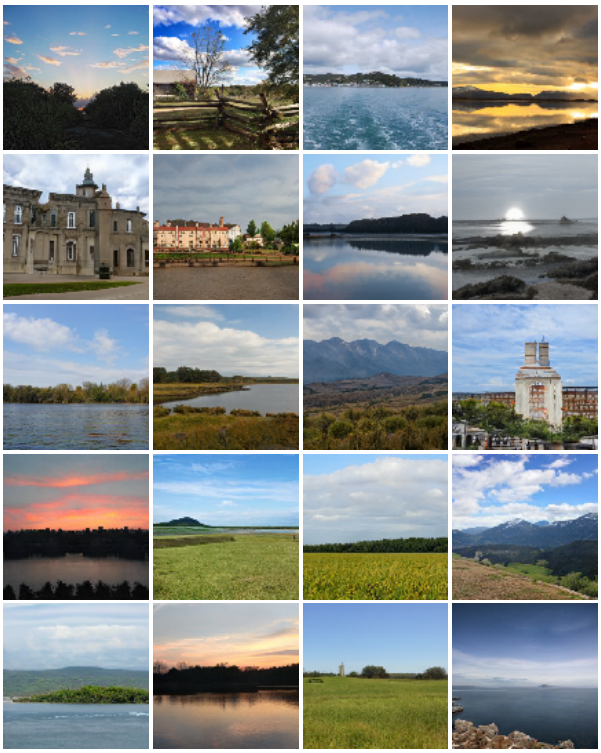
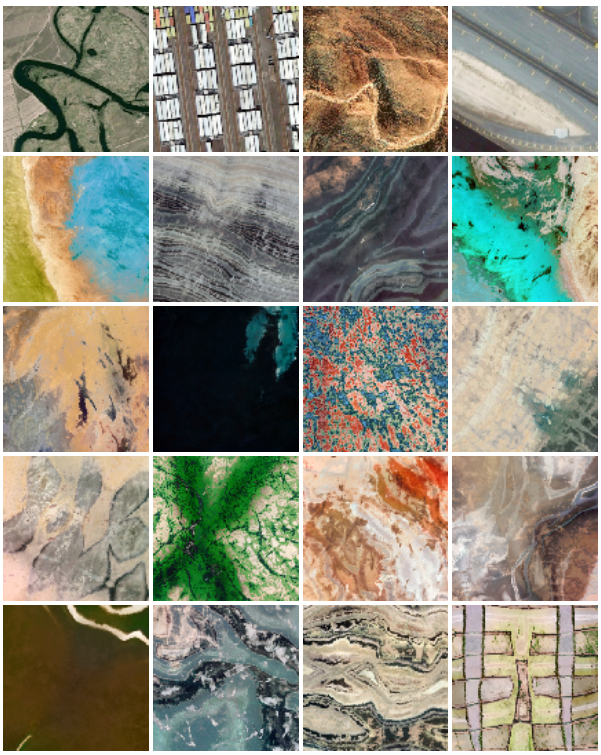


Figure 6: Top: High-resolution samples from CIPS trained on FFHQ-1024 with patch-based training; bottom: samples after training on FFHQ-512 and Landscapes-512. Note that images are JPEG-compressed.

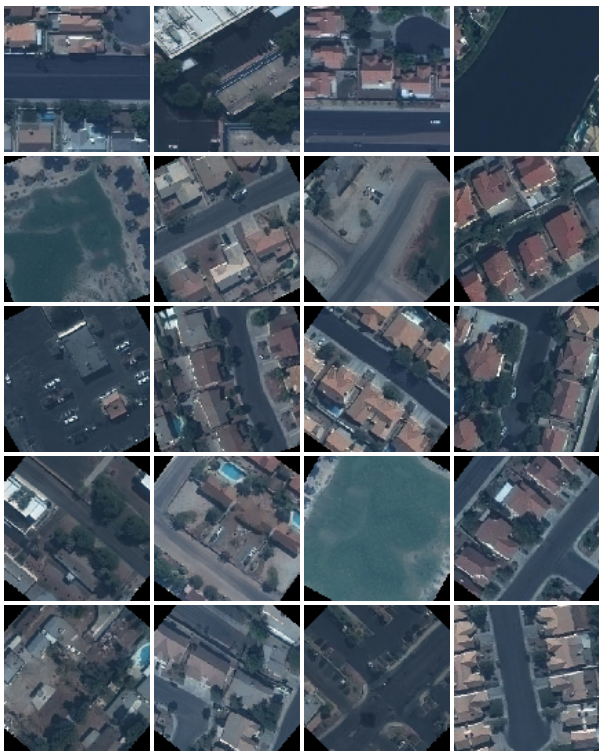
LSUN-Churches



Landscapes



Satellite-Landscapes



Satellite-Buildings

Figure 7: Samples from CIPS generators trained on various datasets. The top row of every grid shows real samples, and the remaining rows contain samples from the models. The samples from CIPS generators are plausible and diverse.

Figure 8: Additional samples of cylindrical panoramas, generated by the CIPS model trained on the Landscapes dataset. The training data contains standard landscape photographs from the Flickr website. No panoramas are provided to the model during training.

Figure 9: Nearest neighbors for generated faces. Within each row, we show a sample from the model on the left. The remaining columns contain real images that are closest to the respective sample in terms of the FaceNet [7] descriptor. The visualization suggests that the CIPS model generalizes well beyond memorization of the training dataset.

Figure 10: Layer-wise style mixing. The two leftmost columns contain source images A and B. In the rightmost three columns, we replace the latent code $\mathbf{w}$ of A with the latent code $\mathbf{w}$ of B at layers (left to right): 6-8, 3-5, 1-2. The visualization suggests that layers 1-2 control the pose and the shape of the head, the middle layers (3-5) control finer geometry such as the shape of eyes, eyebrows and nose, and the final layers (6-8) controls the skin color and the textures. Interestingly, this CIPS model was trained without layerwise mixing, and therefore such decomposition likely arises from the architectural prior.