Arbitrary-Scale Image Synthesis - Supplementary Material

Figure 1. Arbitrary-Scale Image Synthesis on FFHQ with our ScaleParty model. All images were picked randomly and generated without using the truncation trick.

1. Societal Impact

The growth of deepfakes appearing online is a cause for serious concern in a multitude of domains including: politics, non consensual usage of data and the general feeling of losing faith in digital information. GANs are the main technological advancement that enabled the rise of this content. The presented work does not directly lend itself to creation of fake material, in the sense of replacing faces or creating facial expressions based on audio stream. Indirectly though, our method can be used to geometrically manipulate images and in this sense provide malevolent users an additional tool. Efforts in both the US (S. Rept. 116-289 - IDENTIFYING OUTPUTS OF GENERATIVE ADVERSARIAL NETWORKS ACT) and the EU(2021/0106(COD) Artificial Intelligence Act) are aiming to legislate the creation of deepfakes, while private companies try to detect and ban the spreading of deepfake material on the internet. Our method gravitates towards white colored faces in the center of the latent space due to the imbalance on the used data set. There is a clear need to create diverse data sets, where people are represented equally independent of their ability to access technological resources. This will enable research to be used in a more wide spectrum of applications across the globe. In terms of the ever increasing computational costs of training deep neural networks, our presented method overcomes the need for creating independent models at each resolution. It can be therefore be used to reduce the required energy by replacing multiple single resolution models with a single scale consistent one.

2. Limitations

We use artificially multi-scale datasets to train ScaleParty. We downsample the images to acquire different scales. Parmar *et al*. [4] argue that different resizing libraries and methods can have drastic effects on the quality of the resized images. This is an aspect we have not investigated. In equation (5) of the main paper we assume that transitive closure applies to resizing, e.g. resizing from 512×512 to 256×256 is equivalent to resizing to 384×384 as an intermediate step. While this assumption is not true, it still helps us with our scale-consistency objective. Nevertheless, an analysis on a naturally multi-scale dataset would greatly benefit the conclusions of this work.

3. The generator's architecture

In Fig. 2 we can see a schematic of ScaleParty's generator.

4. Multi-scale training policies

In this section we discuss the different scale training policies for FFHQ [3] that we and the methods we compare with deploy:

- CIPS [1] is trained with one target scale: 256.
- MS-PE [2] is trained for 256, 320, 384, 448, 512.
- MSPIE [5] is trained for 256, 384, 512. We use the version of MSPIE with cartesian spatial grid encodings, as it performs the best in terms of FID. Other encoding configurations exhibit similar behavior in the scales they were not trained for.
- We deploy the same setting as MSPIE for ScaleParty-noSC/Full and train for 256, 384 and 512.
- Our ScaleParty-Full which is trained with the scale consistency objective is trained with output resolutions of 256 and 384, but it can perform well even in higher resolutions.
- Our ScaleParty is also trained with output resolutions of 256 and 384. However, in contrast to all aforementioned approaches this is trained for partial generation; the generator is tasked to synthesized a multitude of scales. For example, during training it is generating 384 pixel parts of a 512 resolution full face picture.

For LSUN Datasets [6], we trained MSPIE, ScaleParty-noSC/Full and ScaleParty with outputs of 128 and 192.

In order to facilitate faster and efficient training, we train our scale consistent versions of ScaleParty by continuing from an earlier checkpoint of the ScaleParty-noSC/Full version.

5. Visual Results

In this section we show a qualitative comparison between the state-of-the-art methods and different versions of ScaleParty.

FFHQ [3]: In Fig. 3 we can see visual results of the pretrained models of CIPS [1], MS-PE [2] and [5]. In Fig. 4 we can see the results for ScaleParty-noSC/Full, ScaleParty-Full and ScaleParty. While FID is lower for most scales for the versions trained with only full images, we can observe that the network applies a peculiar effect on the eyes of the faces it generates in scales it did not train for. We can see that both applying the scale consistency objective and partial generation is important for achieving consistent synthesis in arbitrary scales.

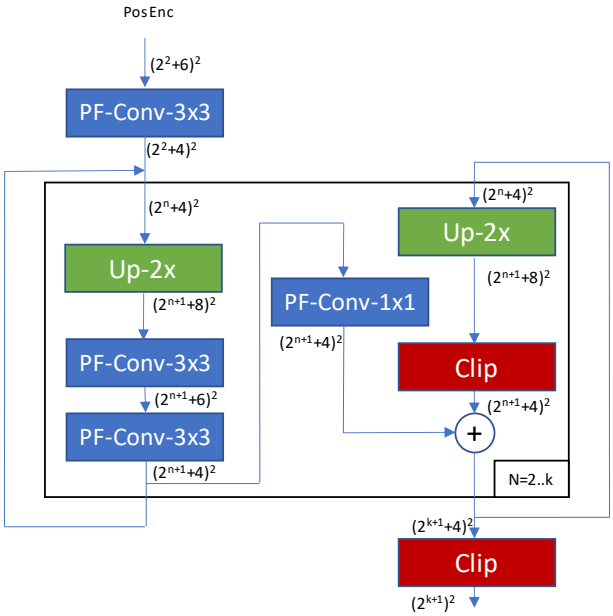

Figure 2. Our generator's architecture. The blue blocks show the padding-free style modulated convolutions. The green blocks denote the operations of bilinear upsampling without aligning corner. The red blocks show the removal of the excess pixels introduced by the input padding in order for the feature maps and the output to match.

Moreover, in Fig. 1 we can see images synthesized at arbitrary scales. As the generator can only output certain resolutions, for scales between them, we generate at a higher resolution and crop the relevant part.

LSUN [6]: In Fig. 5 we can see the qualitative results of MSPIE [5], ScaleParty-noSC/Full and ScaleParty trained for LSUN Bedroom and Church datasets. Note, that in combination with the weaker positional prior that these datasets have compared to FFHQ, we further augment this disparity by applying random cropping as a preprocessing step. Compared with FFHQ, MSPIE is generating more coherent results in the intermediate scale. However, in the case of LSUN Bedroom we can observe that the results are not consistent among different scales.

In Fig. 5, we visualize multiple syntheses of the same latent code and scale but with resampling the injected noise. We observe that ScaleParty is the most consistent among runs, while for MSPIE trained for LSUN Bedroom, we see that noise affects the generated images structurally.

6. UI tool for guided generation

We developed an interactive graphical user interface that permits the user to change the location, zoom factor, size of

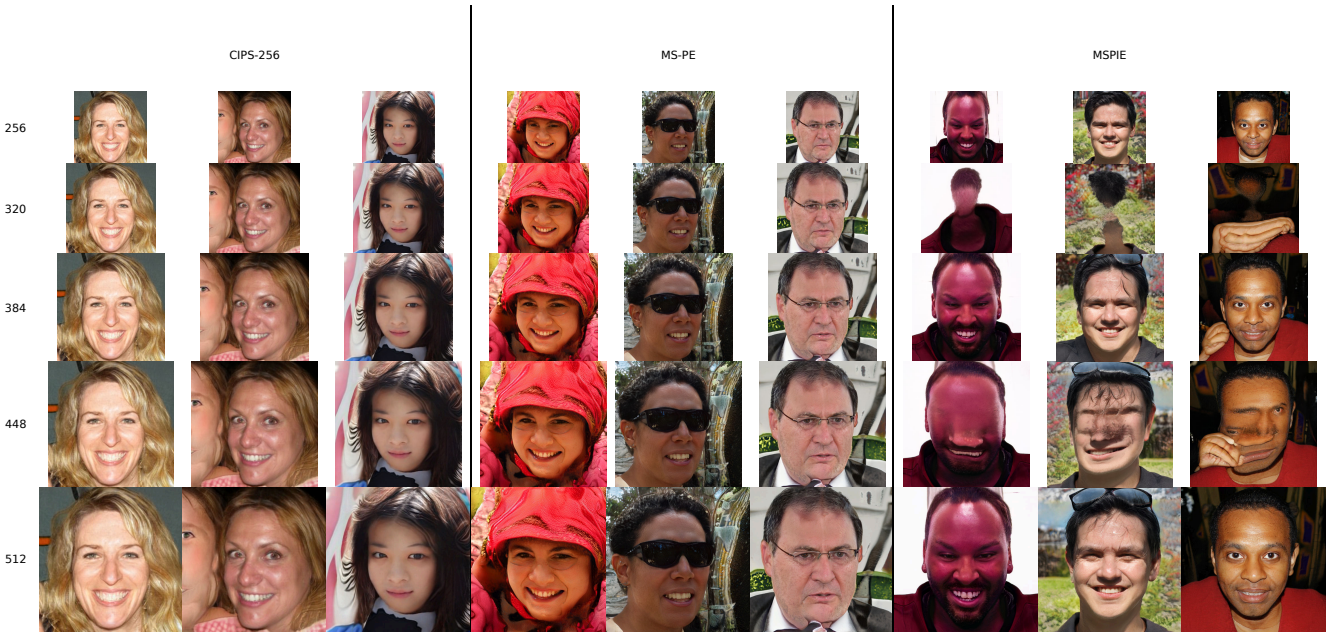

Figure 3. Qualitative results of state-of-the-art methods on FFHQ [3]. All images were picked randomly and generated without using the truncation trick. We find that the generated results from CIPS [1] and MS-PE [2] exhibit a lot more artifacts than MS-PIE and our methods. However, note that MSPIE, while it performs the best in terms of FID among all methods, it is unable to generate in scales it was not trained for and it is the least consistent between the scales it generates.

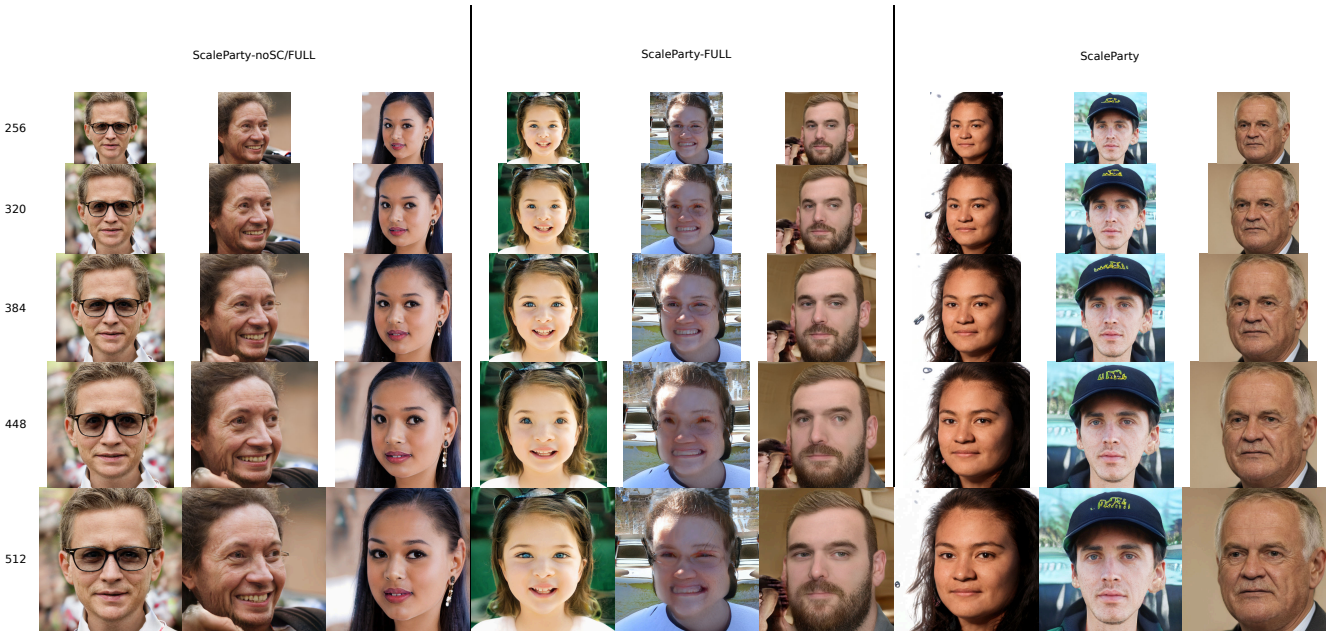

Figure 4. Qualitative results of different versions of our method on FFHQ [3]. All images were picked randomly and generated without using the truncation trick. We note that by drawing more samples the amount of generated images by our models that exhibit visual artifacts is comparable with MSPIE [5] for the scales it was trained for, as it is also supported in the FID calculation. For our methods, we observe that only ScaleParty is able to generate results that are consistent, even for arbitrary scales.

input, aspect ratio and warping of the positional encodings to guide the generation. Please refer to the accompanied video for more details. The tool will be available along with the code and the pretrained models.

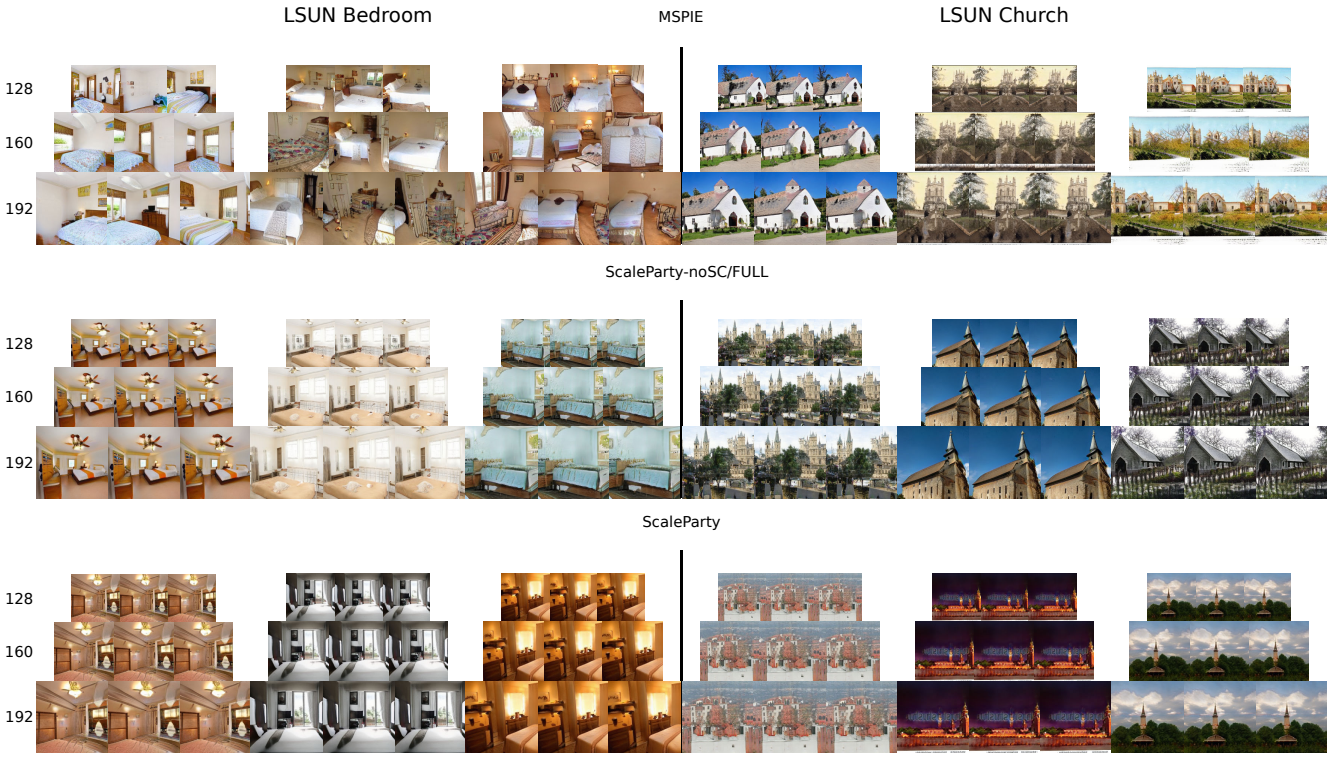

Figure 5. Qualitative results on LSUN Bedroom and Church datasets. All images were picked randomly and generated without using the truncation trick.

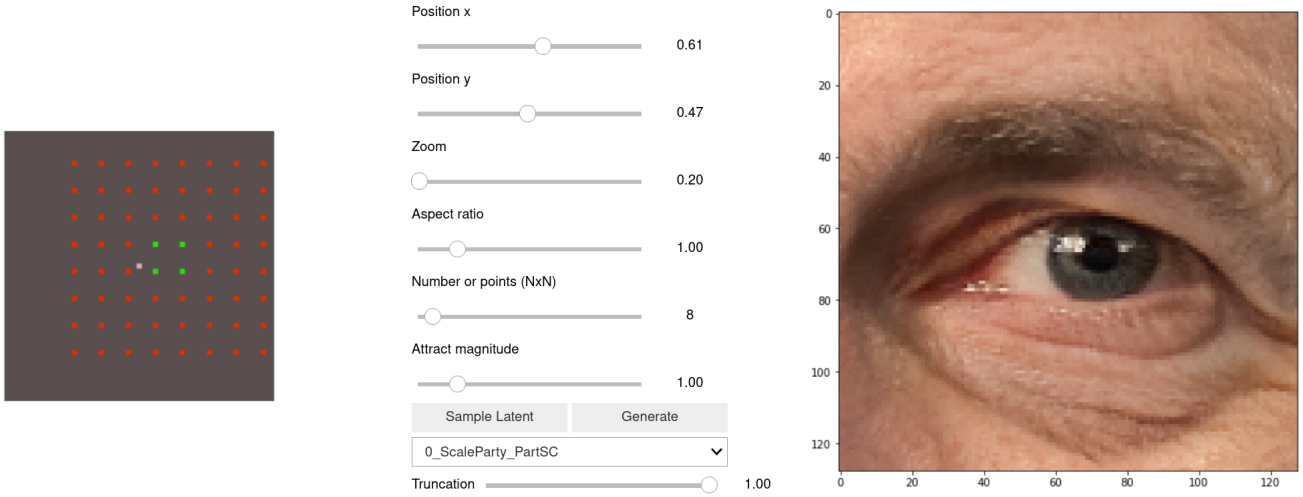

Figure 6. The interface we developed to geometrically manipulate the generation by applying transformations to the positional encodings.

In Fig. 7 we showcase how changing the position, number and layout of the positional encodings can affect the generated image.

References

[1] Ivan Anokhin, Kirill Demochkin, Taras Khakhulin, Gleb Sterkin, Victor Lempitsky, and Denis Korzhenkov. Image generators with conditionally-independent pixel synthesis. *arXiv preprint arXiv:2011.13775*, 2020. 2, 3

[2] Jooyoung Choi, Jungbeom Lee, Yonghyun Jeong, and Sun-

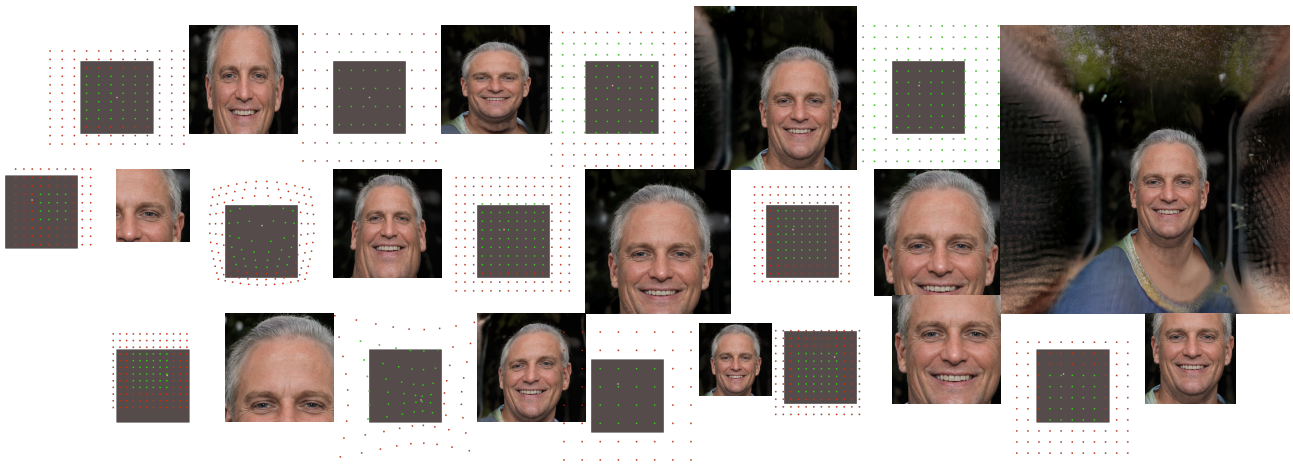

Figure 7. Using our tool, we generate various images using the same latent code. The generated images are connected on their upper left corner with the positional encodings used to guide them. Changing the layout of the input yields different scales, resolutions and transformations. The gray box indicates the area the full face should occupy. The green dots show the actual area of the image space that is generated, while the red ones indicate the positional padding of the input, we utilize to counter the shrinking effect of padding-free convolutions.

groh Yoon. Toward spatially unbiased generative models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14253–14262, October 2021. 2, 3

[3] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3

[4] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On buggy resizing libraries and surprising subtleties in fid calculation. *arXiv preprint arXiv:2104.11222*, 2021. 1

[5] Rui Xu, Xintao Wang, Kai Chen, Bolei Zhou, and Chen Change Loy. Positional encoding as spatial inductive bias in gans. In *arxiv*, December 2020. 2, 3

[6] 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. 2