

Supplementary Material

One Model to Reconstruct Them All: A Novel Way to Use the Stochastic Noise in StyleGAN

Christian Bartz*
christian.bartz@hpi.de

Joseph Bethge*
joseph.bethge@hpi.de

Haojin Yang
haojin.yang@hpi.de

Christoph Meinel
christoph.meinel@hpi.de

Hasso Plattner Institute
University of Potsdam
Potsdam, Germany

* Equal contribution.

1 Further Implementation Details

We implemented our model in PyTorch, as stated in the main paper. We based our implementation of the StyleGAN 1 and StyleGAN 2 models on freely on Github available re-implementations of StyleGAN 1¹ and StyleGAN 2². Our code is also available on Github³ and our training logs can be viewed online⁴.

We perform our experiments on a range of different GPUs with at least 11 GB of GPU memory. For all of our experiments, we train a model for 100 000 iterations, using two GPUs with a batch size of 4 per GPU. We use the Adam [1] optimizer with a cosine annealing learning rate schedule [2] and an initial learning rate of 0.0001. During the preprocessing, all input images are resized to 256×256 pixels, disregarding aspect ratios.

Network Details Our encoder is based on the ResNet architecture, but does not follow the layout of other well-known ResNet feature extractors, *e.g.* ResNet-18, or ResNet-152. Instead, the number of convolutional layers in our feature extractors depends on the resolution of the input image. The number of necessary ResNet blocks can be calculated using the following formula:

$$\text{number of blocks} = 2 + 2 \cdot (\log_2(\text{insize}) - \log_2(\text{outsize})). \quad (1)$$

We first start with two "start blocks", then we use two ResNet blocks for each resolution from input size to output size of the encoder. The output size of the encoder is typically set to 4, whereas we use 256 as our input size. When using 256 and 4 as an input and output size, respectively, we get the following number of ResNet blocks:

$$\text{number of blocks} = 2 + 2 \cdot (\log_2(256) - \log_2(4)) \quad (2)$$

$$= 2 + 2 \cdot (8 - 2) \quad (3)$$

$$= 2 + 2 \cdot 6 \quad (4)$$

$$= 14. \quad (5)$$

Following each ResNet block, the network splits into three branches. If we use StyleGAN 1 as decoder, we only split after the first ResNet block of each resolution. The first split is followed by a global average pooling and a 1×1 convolution that predicts parts of the latent code. The second split is followed by a 1×1 convolution that is used to predict the noise inputs for the current feature map resolution. The third branch goes to the next ResNet block. Please see Figure 2 of the main paper for a structural overview.

2 Reconstruction Results on Pre-Trained StyleGAN Models

Besides a reconstruction performance comparison to other state-of-the-art models, we provide reconstruction results on a range of pre-trained generator models. First, we provide reconstruction results when using a StyleGAN model pre-trained on the FFHQ [10] dataset. In this line of experiments, we trained our encoder on several different datasets and apply encoder and decoder on a range of datasets reporting the reconstruction results in Table 1. Furthermore, we trained StyleGAN generators on the LSUN Church and LSUN Cat datasets [9] and provide the same range of experiments in Table 2 and Table 3. The results of these experiments show that reconstruction with our proposed encoder works using a diverse range of models, thus confirming our hypothesis that our encoder model generalizes very well to various image domains.

Further Visualizations Explaining the Role of Noise

In Figures 2, 3, 4, and 5 we show more detailed results of our noise shifting experiments. These experiments show that noise is used in several ways: (1) Noise can be used to control the content of the generated image. (2) Noise can be used to control the colors of individual pixels of the generated image. (3) A StyleGAN model trained on one dataset can make use of noise in a different way than a StyleGAN model trained on a different dataset, *i.e.* the color coding depends on the StyleGAN model and used dataset for training the StyleGAN model. (4) When only using stochastic noise as input, noise does not have as much influence as when using controlled noise as input.

Further Visualizations Showing Interpolation Results

In Figure 6 we show further interpolation results. Again, we can observe that models trained with the two network approach provide more meaningful interpolations. We can also make

the following observations: (1) If using a model trained on another domain than the input image, *e.g.* a model trained to reconstruct FFHQ using the two network strategy and using images from the LSUN church dataset (see [Figure 6\(b\)](#)), we can see that the encoder explicitly learns to embed the latent code into the regions where faces can be generated, regardless of the input image. We can hence conclude that the encoder does not learn anything about the content of the image when embedding into the latent code. (2) in [Figure 6\(b\)](#) and [Figure 6\(d\)](#), we can observe that the predicted noise maps are “rendered” on top of the results of the latent code. This might be because the two networks in our two network architecture are independent of each other. In the future it might be worthwhile to achieve a closer coupling of noise and latent code in the two network approach. (3) Overall, we can observe that, when not training with the two network strategy, the latent code is only used to add color variations into the resulting images. This can, for instance, be observed in [Figure 6\(a\)](#) in the second-to-right image in the bottom-most row. Here, we can see that the latent code also influences the colors of the resulting image, since the image on the right is the reconstruction of the second image without any interpolation between the two input images.

Besides training two independent networks to retain more semantic information in the latent code, we found that such results are also possible with careful tuning of the learning rate and a two-step training strategy. We denote this approach as the learning rate strategy. Here, we first train the model without learning the layers that project into the noise inputs. Later, we fine-tune the model including the noise layers but we reduce the learning rate of the layers responsible for predicting the latent code to avoid them to forget everything they learned because the noise inputs are simpler to optimize. We compare the two approaches for maximising the semantic meaning of the latent code, by providing a comparison of interpolations for the approaches based on the two network and the learning rate strategy in [Figure 7](#). We can observe that the learning rate split strategy produces better qualitative results. However, it is very difficult to achieve these results for each StyleGAN version, since the learning rate has to be tuned for each model individually.

3 Quantitative Reconstruction Results with Two Stage Training and Two Networks

In [Table 4](#) we show quantitative results for reconstruction on multiple datasets when using a model trained with the two network strategy. The results show that a model trained with the two network strategy is still able to provide meaningful reconstructions. However, the results also show that the resulting reconstructions are of worse quality than the reconstructions that nearly completely rely on the noise input. Furthermore, we can see that the resulting models are not well suited for cross domain reconstruction. We argue that this is because the latent code is specialized to a specific dataset/data distribution and is not able to handle inputs that are different, since the encoder has learned to project the input image into the regions of the latent space for the specific type of data it has been trained for. From these observations we conclude that our models, which are not trained to maximise the semantic meaning of the latent code, are so versatile, because they learn to encode the content of the image in the noise maps and to use the values of each pixel to further encode the color of the pixel, making them independent of the latent code in StyleGAN, which is adjusted for one distribution only. However, it would be interesting to investigate the latent projections of different encoders to learn more about the structure of the latent code in StyleGAN. We leave these experiments

Training Dataset, StyleGAN Version, Projection Target	Dataset and Metric for Evaluation											
	FFHQ			Church			Bedroom			Cat		
	FID \downarrow	PSNR \uparrow	SSIM \uparrow	FID \downarrow	PSNR \uparrow	SSIM \uparrow	FID \downarrow	PSNR \uparrow	SSIM \uparrow	FID \downarrow	PSNR \uparrow	SSIM \uparrow
FFHQ, 1, \mathcal{Z}	9.85	25.03	0.91	7.17	20.37	0.88	3.74	23.32	0.91	4.17	22.14	0.88
FFHQ, 1, \mathcal{W}	0.64	25.54	0.94	1.37	22.26	0.93	0.55	24.21	0.94	0.90	23.02	0.91
FFHQ, 2, \mathcal{Z}	3.92	24.39	0.88	4.66	19.87	0.82	3.24	22.71	0.86	4.47	21.50	0.84
FFHQ, 2, \mathcal{W}	0.75	29.11	0.95	1.23	24.48	0.94	0.57	28.13	0.96	0.96	26.01	0.93
Church, 1, \mathcal{Z}	17.28	19.83	0.86	3.17	23.32	0.91	4.22	21.90	0.90	4.98	20.50	0.86
Church, 1, \mathcal{W}	3.30	20.99	0.90	0.26	26.18	0.95	1.25	23.65	0.94	1.37	22.15	0.90
Church, 2, \mathcal{Z}	12.24	20.92	0.83	3.17	23.08	0.86	9.76	22.04	0.85	7.33	20.76	0.82
Church, 2, \mathcal{W}	2.33	23.10	0.91	0.21	28.99	0.95	0.60	26.43	0.96	1.06	24.35	0.91
Bedroom, 1, \mathcal{Z}	5.82	23.14	0.91	2.23	23.41	0.92	1.13	26.94	0.95	2.11	23.71	0.90
Bedroom, 1, \mathcal{W}	1.60	24.10	0.91	1.16	23.00	0.93	0.30	26.22	0.95	0.79	23.86	0.91
Bedroom, 2, \mathcal{Z}	7.17	22.70	0.87	6.48	20.49	0.84	2.96	24.29	0.88	5.10	21.97	0.85
Bedroom, 2, \mathcal{W}	1.59	26.44	0.93	1.06	26.97	0.94	0.36	31.01	0.97	0.84	26.78	0.93
Cat, 1, \mathcal{Z}	39.65	24.60	0.92	6.68	24.25	0.93	4.01	26.71	0.94	2.96	25.03	0.92
Cat, 1, \mathcal{W}	1.12	24.94	0.93	0.88	24.29	0.94	0.28	26.40	0.96	0.57	24.86	0.93
Cat, 2, \mathcal{Z}	5.31	23.61	0.90	2.83	22.73	0.90	2.18	24.49	0.92	2.71	23.31	0.89
Cat, 2, \mathcal{W}	1.34	28.38	0.95	0.55	27.99	0.95	0.33	30.35	0.97	0.68	28.41	0.94

Table 1: The results of our reconstruction experiments. We use a StyleGAN model for decoding, which was pre-trained on the FFHQ dataset [1] and is not updated during the training of the encoder. Only the first highlighted row uses an encoder which is also trained on FFHQ, the other encoders are trained on different LSUN datasets [2]. The first column shows the dataset, the version of StyleGAN (1, 2), and the projection target (\mathcal{Z} , \mathcal{W}). Each model is evaluated on different datasets and we report FID, PSNR, and SSIM.

open for future work.

Training Dataset, StyleGAN Version, Projection Target	Dataset and Metric for Evaluation											
	FFHQ			Church			Bedroom			Cat		
	FID [↓]	PSNR [↑]	SSIM [↑]	FID [↓]	PSNR [↑]	SSIM [↑]	FID [↓]	PSNR [↑]	SSIM [↑]	FID [↓]	PSNR [↑]	SSIM [↑]
FFHQ, 2, \mathcal{Z}	5.87	25.19	0.91	8.28	19.95	0.84	4.96	23.45	0.89	3.90	22.65	0.87
FFHQ, 2, \mathcal{W}	0.37	29.48	0.96	1.21	23.99	0.93	0.48	27.96	0.96	0.83	25.95	0.93
Church, 2, \mathcal{Z}	13.31	21.04	0.86	2.58	22.92	0.89	4.04	22.85	0.89	5.95	21.56	0.86
Church, 2, \mathcal{W}	2.06	24.18	0.92	0.19	30.12	0.96	0.40	27.73	0.96	0.96	25.40	0.92
Bedroom, 2, \mathcal{Z}	8.05	24.07	0.90	3.05	22.45	0.88	2.28	26.17	0.92	2.85	23.70	0.89
Bedroom, 2, \mathcal{W}	0.75	26.75	0.94	0.62	26.69	0.95	0.17	31.64	0.97	0.53	27.28	0.93
Cat, 2, \mathcal{Z}	10.72	24.31	0.91	4.12	22.23	0.87	4.62	24.87	0.90	3.62	24.42	0.90
Cat, 2, \mathcal{W}	0.92	28.06	0.95	0.48	28.15	0.96	0.25	30.55	0.97	0.48	28.60	0.94

Table 2: The results of further reconstruction experiments. We use a StyleGAN model for decoding, which was pre-trained on the LSUN cat dataset and not updated during the training of the encoder. We train the decoder of our models on different datasets (FFHQ and LSUN datasets), StyleGAN 2, and different projection targets (\mathcal{Z} , \mathcal{W}) as shown in the first column. We evaluate each model on different datasets and report FID, PSNR, and SSIM.

Training Dataset, StyleGAN Version, Projection Target	Dataset and Metric for Evaluation											
	FFHQ			Church			Bedroom			Cat		
	FID [↓]	PSNR [↑]	SSIM [↑]	FID [↓]	PSNR [↑]	SSIM [↑]	FID [↓]	PSNR [↑]	SSIM [↑]	FID [↓]	PSNR [↑]	SSIM [↑]
FFHQ, 2, \mathcal{Z}	2.89	26.52	0.93	2.59	21.75	0.91	1.33	25.41	0.94	1.66	23.77	0.91
FFHQ, 2, \mathcal{W}	0.33	29.32	0.96	0.87	24.29	0.94	0.41	27.77	0.96	0.67	25.88	0.93
Church, 2, \mathcal{Z}	6.75	21.82	0.89	0.85	25.69	0.92	1.92	23.77	0.92	2.52	22.49	0.89
Church, 2, \mathcal{W}	2.05	24.34	0.92	0.18	29.62	0.95	0.44	27.74	0.96	0.88	25.36	0.92
Bedroom, 2, \mathcal{Z}	5.19	24.55	0.91	1.72	23.86	0.92	0.87	27.63	0.94	1.50	24.22	0.90
Bedroom, 2, \mathcal{W}	0.83	26.84	0.94	0.72	26.59	0.95	0.16	31.58	0.97	0.53	27.25	0.93
Cat, 2, \mathcal{Z}	6.33	25.36	0.92	1.58	24.72	0.92	1.16	26.77	0.94	1.58	25.22	0.91
Cat, 2, \mathcal{W}	0.56	28.25	0.95	0.36	28.28	0.96	0.17	30.72	0.97	0.38	28.63	0.94

Table 3: The results of further reconstruction experiments. Here, we use a StyleGAN model for decoding, which was pre-trained on the LSUN church dataset and not updated during the training of the encoder. We train the decoder of our models on different datasets (FFHQ and LSUN datasets), StyleGAN 2, and different projection targets (\mathcal{Z} , \mathcal{W}) as shown in the first column. We evaluate each model on different datasets and report FID, PSNR, and SSIM.



(a) Results for models based on StyleGAN models pre-trained on the LSUN Cat dataset.



(b) Results for models based on StyleGAN models pre-trained on the LSUN Church dataset.

Figure 1: Results of further Reconstruction experiments with StyleGAN models pre-trained on other datasets than FFHQ. Images in the first row are real images, images in the following rows are reconstructions where the naming is as follows: StyleGAN variant, latent projecting strategy. Best viewed in color.

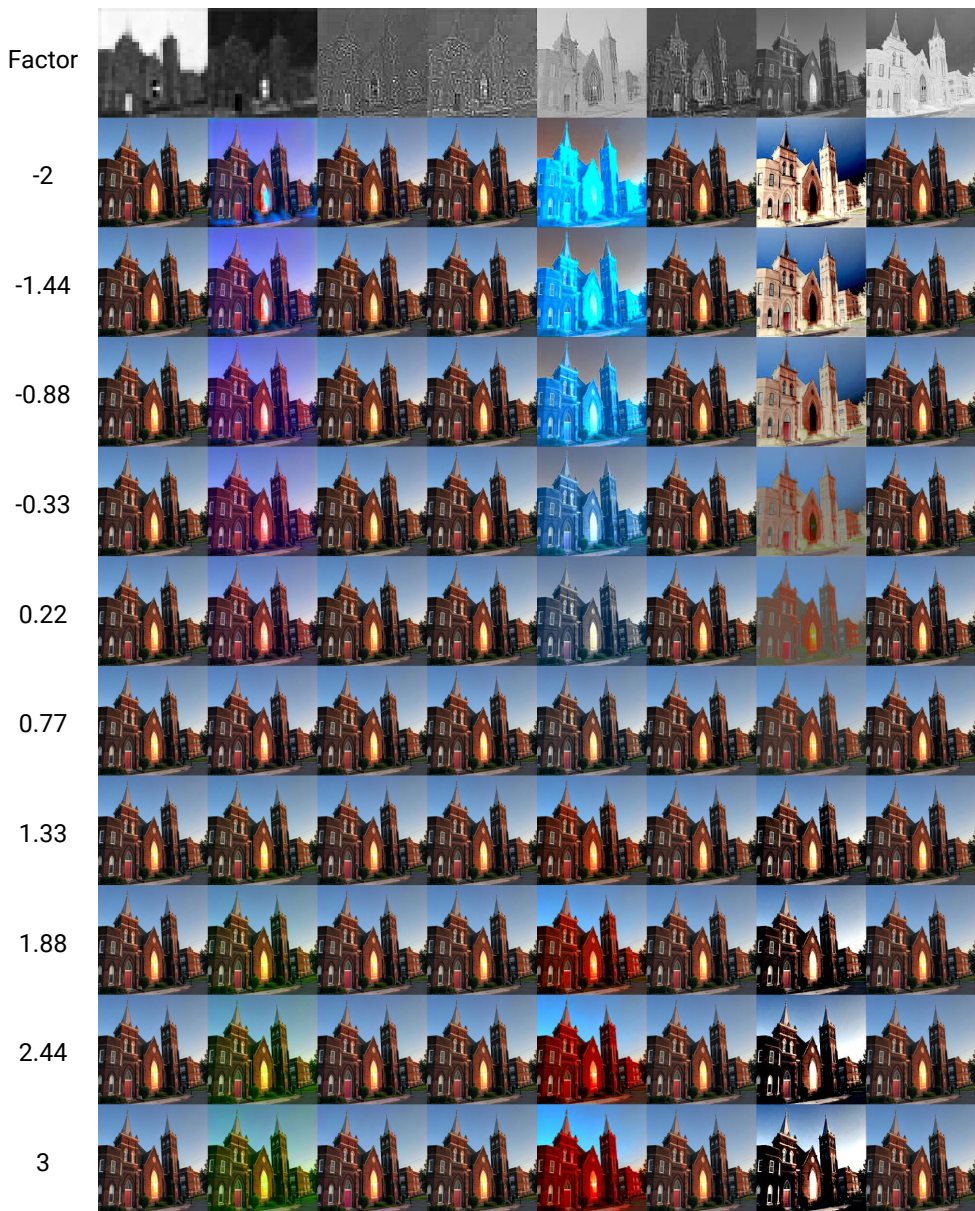


Figure 2: Longer version of noise shifting experiments with a StyleGAN 2 generator pre-trained on the FFHQ dataset. Each column shows the results when “shifting” each pixel of the corresponding noise map shown in the first row of the column by multiplying the noise map with the factors indicated at the left side. It is clearly visible that the noise maps are not only used to encode the content of an image, but that noise can also be used to encode color and contrast of images. However, not all noise maps are necessary for this color coding, as the result does not change for some noise maps, regardless of the factor used.

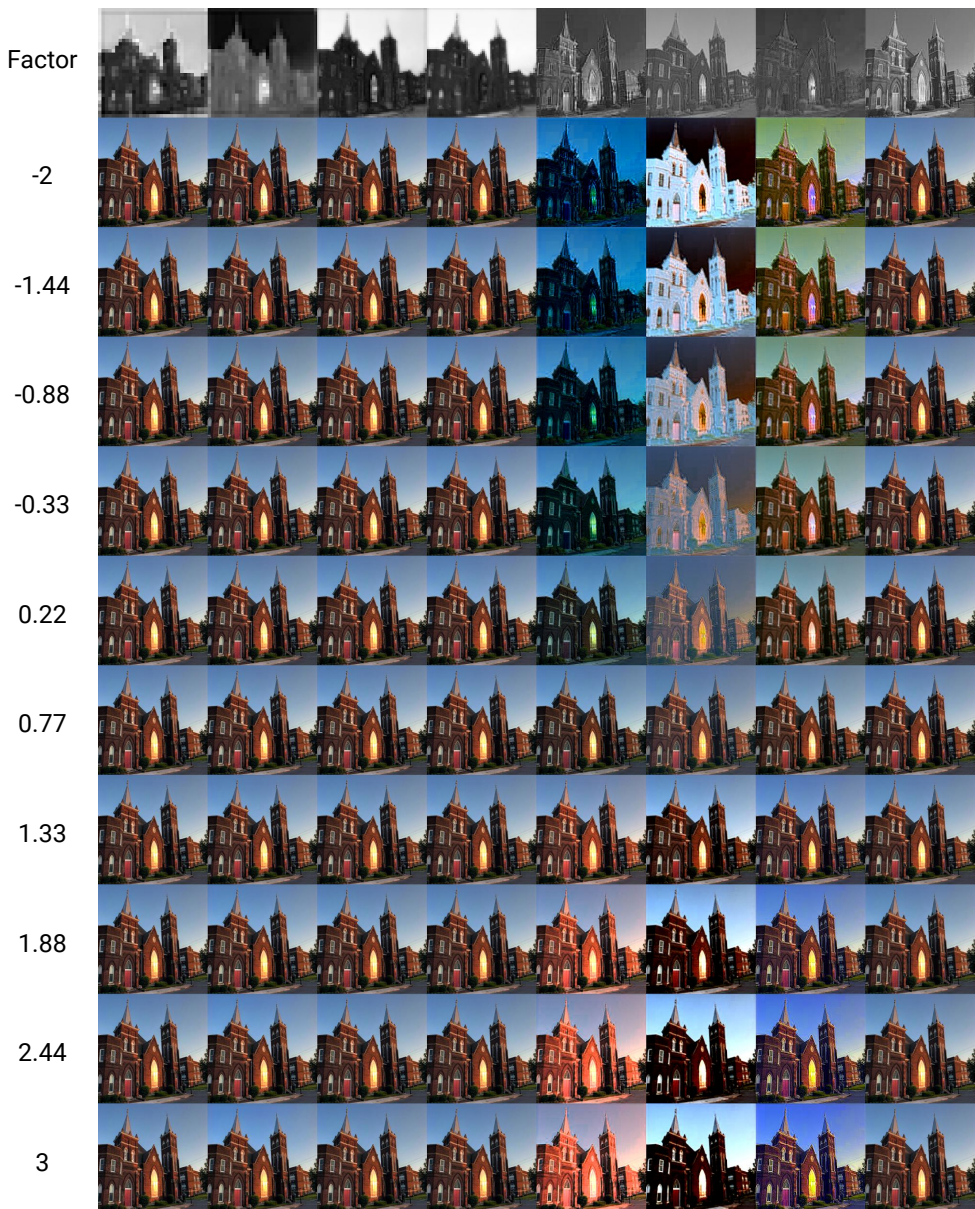


Figure 3: Further results of noise shifting experiments. Here we show the results with a StyleGAN 2 generator pre-trained on the LSUN church dataset. The semantics are the same as in Figure 2. Here it is also clearly visible that the noise maps are not only used to encode the content of an image, but that noise is also used to encode color and contrast of images, but for this model different noise maps are used and the color model encoded is also different.

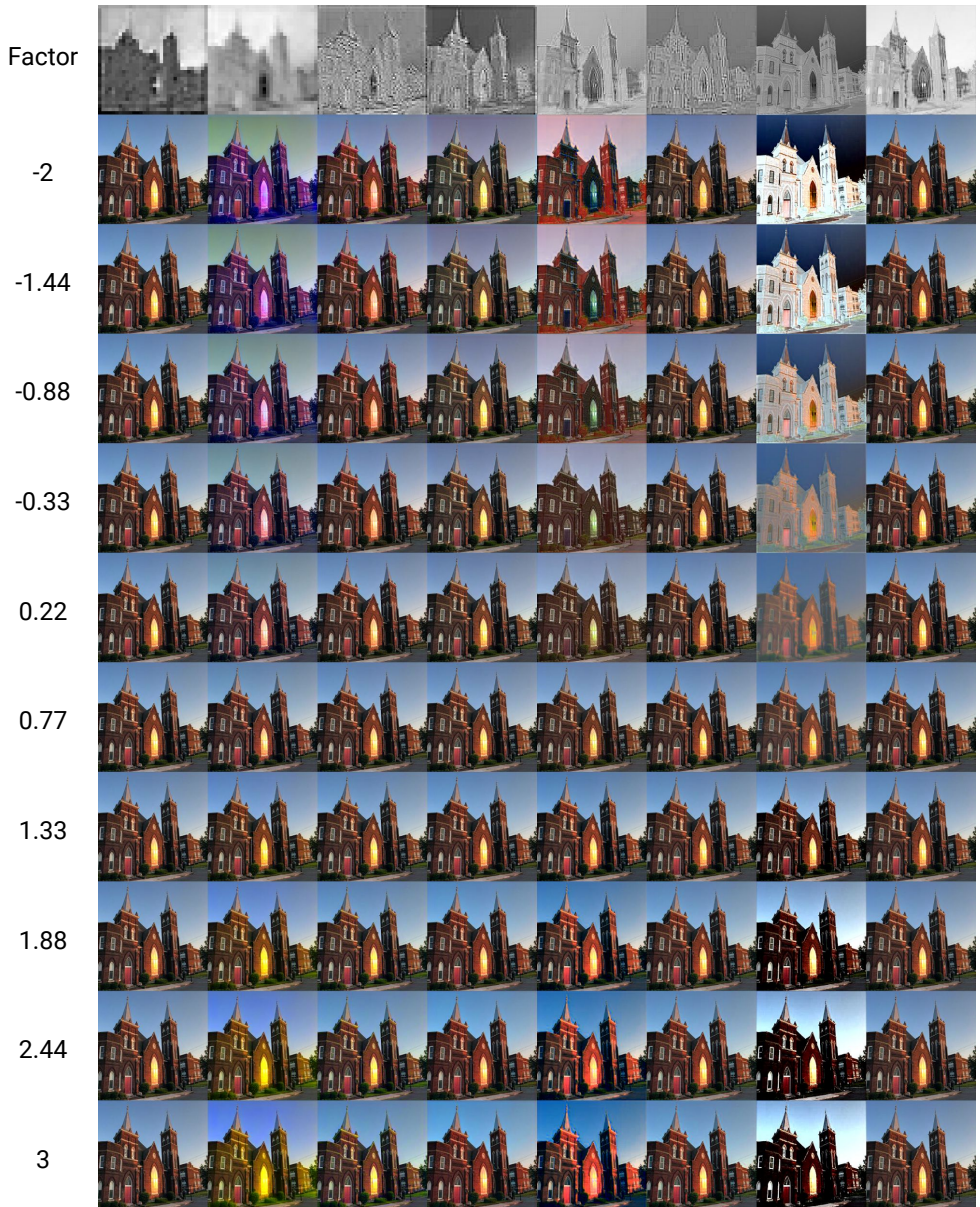


Figure 4: Further results of noise shifting experiments. Here we show the results with a StyleGAN 2 generator pre-trained on the LSUN cat dataset. The semantics are the same as in Figure 2 and Figure 3. Here it is also clearly visible that the noise maps are not only used to encode the content of an image, but that noise is also used to encode color and contrast of images, but for this model different noise maps are used and the color model encoded is again different.

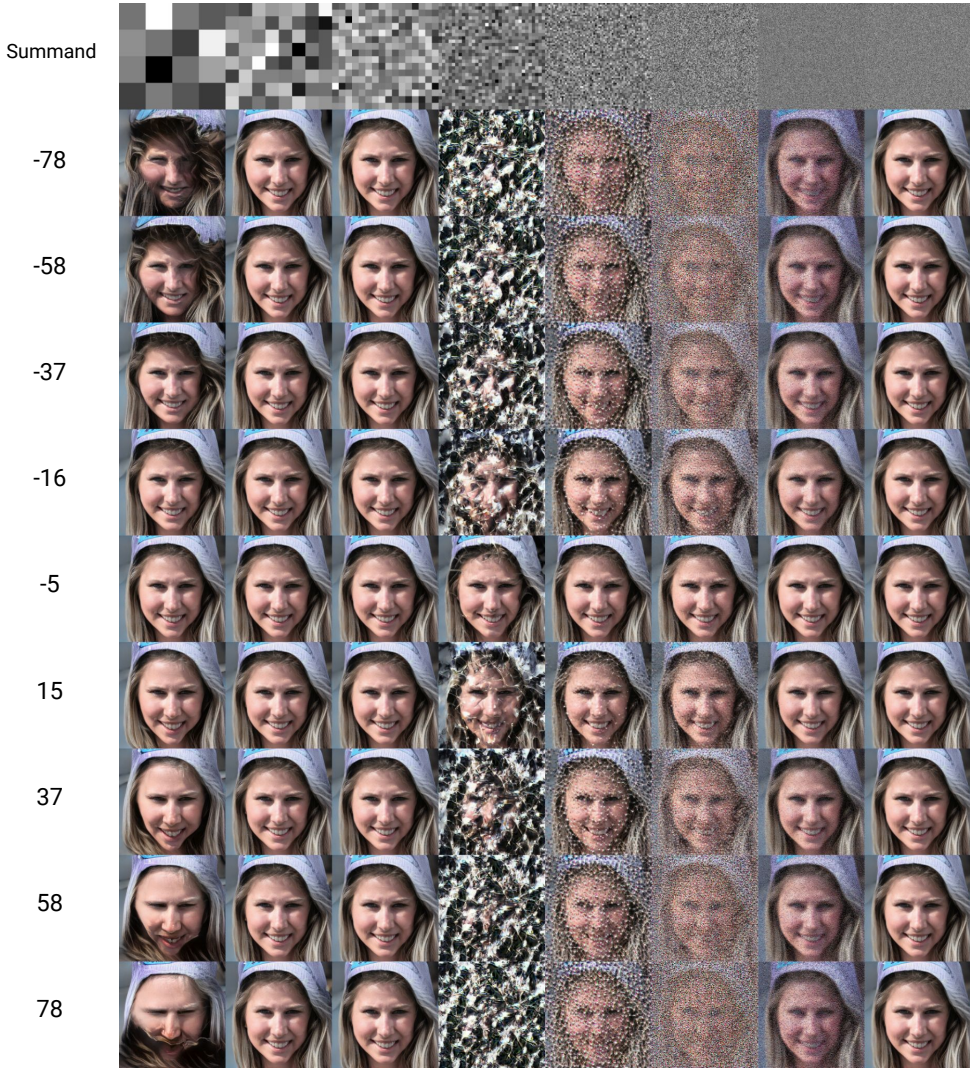


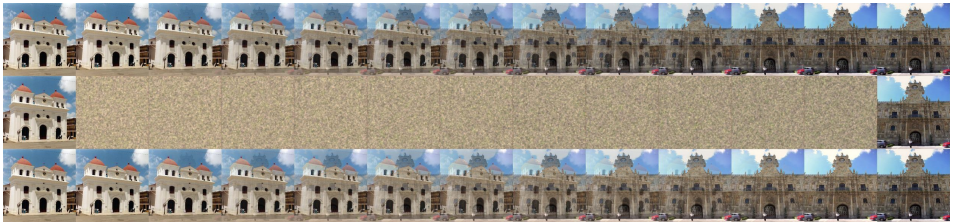
Figure 5: Noise shift experiments when using unconditional image generation with a StyleGAN model trained on the FFHQ dataset. Similar to our other noise shifting experiments, we can see that the shifted noise has influence on the resulting image. However, the influence of the noise highly depends on the layer the noise is applied to. We can see that at some parts the color of image parts change based on the noise. This is in line with our other experiments but since the noise inputs do not encode the content of the image noise is added to the generated image. Here, we did not multiply the noise inputs, we added a constant amount to each pixel.



(a) Encoder trained on FFHQ, decoder pre-trained on FFHQ, with StyleGAN 2, and projecting into \mathcal{W} .



(b) Encoder trained on FFHQ, decoder pre-trained on FFHQ, with StyleGAN 2, projecting into \mathcal{W} , and our two network strategy.



(c) Encoder trained on LSUN cat, decoder pre-trained on FFHQ, with StyleGAN 2, and projecting into \mathcal{W} .



(d) Encoder trained on LSUN cat, decoder pre-trained on FFHQ, with StyleGAN 2, projecting into \mathcal{W} , and our two network strategy.

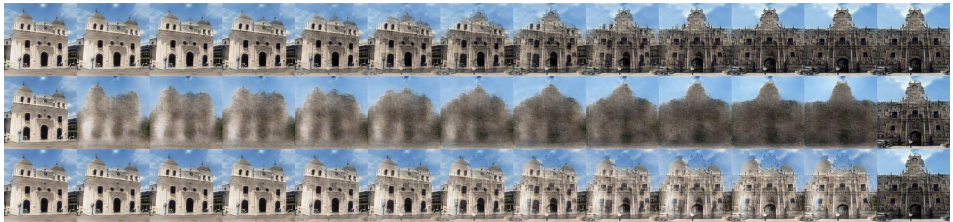
Figure 6: Several images showcasing the behaviour of our models when interpolating latent code and noise maps between two input images. We can observe that models trained without the two network strategy do not make any "intelligent" use of the latent code. The latent code is only used to encode some colors, as can be seen in the second-to-right image in the bottom-most row in (a) and (c). For the other models, we can observe that our two network strategy can only be used to faithfully reconstruct images of the same data distribution. A semantically meaningful interpolation can be observed in (b), but since the base model was trained on the FFHQ dataset, the latent code can only directly be used to generate faces. The results in (d) are similar to the results in the main paper, but here we can also see that noise is only rendered on top of the image generated by the latent code.



(a) Encoder trained on LSUN church, decoder pre-trained on FFHQ, with StyleGAN 1, and projecting into \mathcal{W} .



(b) Encoder trained on LSUN church, decoder pre-trained on FFHQ, with StyleGAN 1, projecting into \mathcal{W} , and our two network strategy.



(c) Encoder trained on LSUN church, decoder pre-trained on FFHQ, with StyleGAN 1, projecting into \mathcal{W} , and using the learning rate strategy.

Figure 7: Further images showcasing the behavior of our models when interpolating latent code and noise maps between two input images. Here, we used models trained on StyleGAN 1 that project into \mathcal{W} . We compare plain projection into \mathcal{W} , our two network strategy and our learning rate strategy to maximise the semantic meaning of the predicted latent code. We can observe that the results of the two network and the learning rate strategy (see (b) and (c)) are of similar visual quality. However, the reconstructions in (c) have a slightly better visual quality and also the interpolations seem to be more reasonable. We conclude that the learning rate strategy is superior to the two network strategy, but it is more difficult to find the correct learning rate ratio, as already discussed in the main paper.

Training Dataset, StyleGAN Version, Projection Target	Dataset and Metric for Evaluation											
	FFHQ			Church			Bedroom			Cat		
	FID \downarrow	PSNR \uparrow	SSIM \uparrow	FID \downarrow	PSNR \uparrow	SSIM \uparrow	FID \downarrow	PSNR \uparrow	SSIM \uparrow	FID \downarrow	PSNR \uparrow	SSIM \uparrow
FFHQ, 2, \mathcal{Z}	24.27	15.35	0.69	38.62	12.76	0.60	41.53	14.15	0.66	29.47	13.77	0.63
FFHQ, 2, \mathcal{W}	9.73	22.16	0.85	22.53	17.33	0.74	20.35	19.09	0.80	15.20	18.92	0.79
Church, 2, \mathcal{Z}	31.02	16.32	0.72	18.50	17.78	0.74	27.47	17.48	0.75	23.33	16.35	0.71
Church, 2, \mathcal{W}	36.19	18.81	0.79	5.46	21.03	0.84	13.94	19.21	0.81	15.77	18.50	0.78
Bedroom, 2, \mathcal{Z}	29.12	15.94	0.68	23.61	15.14	0.66	15.72	17.00	0.70	24.38	15.53	0.66
Bedroom, 2, \mathcal{W}	16.12	21.23	0.83	15.23	19.46	0.80	6.67	22.17	0.84	10.34	20.76	0.82
Cat, 2, \mathcal{Z}	23.55	19.03	0.78	22.14	17.25	0.74	20.52	19.11	0.78	18.81	18.24	0.76
Cat, 2, \mathcal{W}	20.22	21.75	0.86	15.73	19.64	0.83	10.09	21.30	0.85	9.00	21.19	0.85

Table 4: Image reconstruction results for models trained with our two network strategy. We can observe that these models are not as versatile as our other image reconstruction models that are not trained to maximise the semantic meaning of the latent code. However, the reconstruction quality is still high on the datasets they have been trained on.

References

- [1] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 4
- [2] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015. 1
- [3] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR*, 2017. 1
- [4] 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, 4