Conditional Image Sampling by Deep Automodulators

Ari Heljakka[1,2] **Yuxin Hou**[1] **Juho Kannala**[1] **Arno Solin**[1]
[1]Aalto University, Espoo, Finland
[2]GenMind Ltd
`firstname.lastname@aalto.fi`

Abstract

We introduce a novel autoencoder model that deviates from traditional autoencoders by using the full latent vector to modulate each layer of the decoder independently of each other, taking advantage of adaptive instance normalization. Further, we demonstrate how such an 'automodulator' specifically allows for a novel principled method to enforce latent space disentanglement. We do this by passing random combinations of latents through the network more than once, whilst maintaining consistency between multiple passes. This conserves the layer-specific information in latent codes. As the backbone, we extend the recent PIONEER model which retains input identity and image sharpness in higher resolutions than plain VAEs. As a first application, we demonstrate conditional sampling of realistic face images by having some decoder layers fixed to the latent code of a 'driving' input image.

1 Introduction

In image generation, the probability mass representing sensible images (such as human faces) lies concentrated on an effectively low-dimensional manifold. Even if impressive results have been shown for image generation (*e.g.*, by GANs [2, 10]), efficient reconstruction, sampling, or manipulation remain open problems. Deep generative autoencoders provide a principled approach for automatic feature extraction and modelling. They are typically set up as single-pass encoder–decoder structures, where a sample enters from one end and is reconstructed at the other. It would appear that the reconstructed samples could be re-introduced to the encoder, repeating the process, and requiring consistency between the passes. However, especially in variational autoencoders (VAEs, [8, 11]), the reconstructed sample is not guaranteed to be identical to the input even at convergence, due to the use of the reparametrization trick. This makes the idea of multiple passes questionable for VAEs, compounded by their well-known empirical tendency to blur images. However, in models without reparametrization, such as AGE [13], the individual sample could in principle be conserved when doing more than a single pass, and recently such models have been shown to retain input identity and image sharpness in high resolutions (see [3, 4]). This would allow for measuring consistency between the 1st and 2nd pass at any network layer. Furthermore, by utilizing adaptive instance normalization (AdaIn, [7]) in the decoder, we can use the latent codes to *modulate* the statistics of each decoder layer separately. This allows us to mix the latent codes of separate samples and measure the conservation of layer-specific information for each, which enforces disentanglement of layer-specific properties.

We introduce such a model and turn this idea into a novel way of formulating reconstruction losses that improve disentanglement of features in latent space. First, we provide a clean model (called *automodulator*) that can be used as a simple autoencoder (with no 'discriminators' or *ad hoc* additions) but it produces sharp outputs unlike typical VAE models. Second, it provides the same style mixing capabilities as StyleGAN [10] but, critically, our model can also directly operate on new *real* image

4th workshop on Bayesian Deep Learning (NeurIPS 2019), Vancouver, Canada.

inputs. Third, our model is completely general-purpose, and allows not only style mixing but also latent space manipulation such as morphing the image by latent space interpolation.

2 Models and Methods

Our interest is in unsupervised training of an autoencoder wherein the inputs $\mathbf{x}$ are images fed through an encoder $\boldsymbol{\phi}$ to form a low-dimensional latent space representation $\mathbf{z}$ (we use $\mathbf{z} \in \mathbb{R}^{512}$, normalized to unity). This representation can then be decoded back into an image $\hat{\mathbf{x}}$ through a decoder $\boldsymbol{\theta}$. For stable training, we adopt the progressively growing architecture of Balanced PIONEER [3, 5]. The convolutional layers of the symmetric encoder and decoder are faded in gradually during the training, in tandem with the resolutions of training images and generated images.

Adaptive Instance Normalization (AdaIn) A traditional decoder architecture would start from a small-resolution image and expand it layer by layer until the full image is formed, feeding the full information of the latent code via the decoder layers. In contrast, our decoder is composed of layer-wise functions $\boldsymbol{\theta}_i(\boldsymbol{\xi}_{i-1}, \mathbf{z})$ that separately take a 'canvas' variable $\boldsymbol{\xi}_{i-1}$ denoting the content input from the preceding decoder layer (see Fig. 1a), and the actual (shared) latent code $\mathbf{z}$. First, each deconvolutional layer #i computes its feature map output $\boldsymbol{\chi}_i$ from $\boldsymbol{\xi}_{i-1}$ as in traditional decoders. Second, we take the AdaIn normalization step that uses $\mathbf{z}$ to modulate the channel-wise mean μ and variance σ^2 of the output. To do this, we need a map $g_i : \mathbf{z} \mapsto (\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2)$, arriving at:

$$\boldsymbol{\xi}_i = \mathrm{AdaIn}(\boldsymbol{\chi}_i, g_i(\mathbf{z})) = \boldsymbol{\sigma}_i \left(\frac{\boldsymbol{\chi}_i - \mu(\boldsymbol{\chi}_i)}{\sigma(\boldsymbol{\chi}_i)} \right) + \boldsymbol{\mu}_i. \tag{1}$$

We implement g_i for layer #i as a fully connected linear layer, with output size $2\,C_i$ for C_i channels. Layer #1 starts from a constant input $\boldsymbol{\xi}^{(0)} \in \mathbb{R}^{4\times4}$. Here, we focus on pyramidal decoders with monotonically increasing resolution and decreasing number of channels, but any deep decoder would be applicable. The AdaIn-based architecture allows the output of layer #i to be not solely determined by the input from layer #$i-1$, enabling finer control over the image information and image mixing schemes.

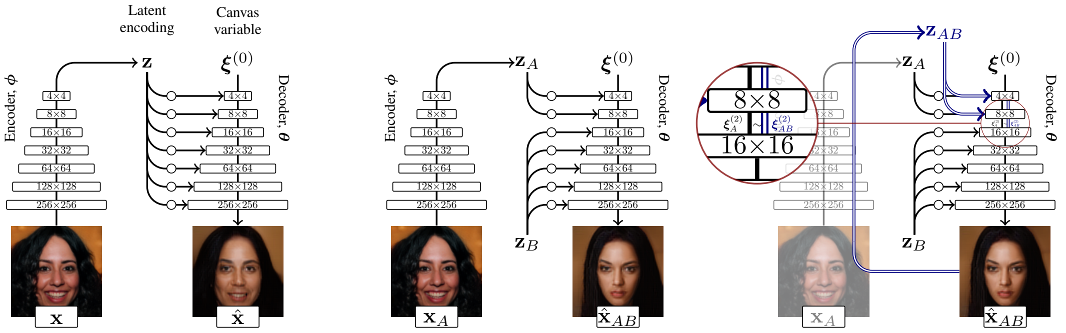
Layer-specific Loss Metrics In training of regular AGE and PIONEER models, the encoder $\boldsymbol{\phi}$ and the decoder θ are trained at separate steps, optimizing either the loss $\mathcal{L}_\phi$ or $\mathcal{L}_\theta$, correspondingly:

$$\begin{aligned}
\mathcal{L}_\phi &= \mathrm{D}_{\mathrm{KL}}[q_\phi(\mathbf{z} \,|\, \mathbf{x}) \,\|\, \mathrm{N}(\mathbf{0}, \mathbf{I})] - \mathrm{D}_{\mathrm{KL}}[q_\phi(\mathbf{z} \,|\, \hat{\mathbf{x}}) \,\|\, \mathrm{N}(\mathbf{0}, \mathbf{I})] + \lambda_{\mathcal{X}}\, d_{\mathcal{X}}(\mathbf{x}, \boldsymbol{\theta}(\boldsymbol{\phi}(\mathbf{x}))), \\
\mathcal{L}_\theta &= \mathrm{D}_{\mathrm{KL}}[q_\phi(\mathbf{z} \,|\, \hat{\mathbf{x}}) \,\|\, \mathrm{N}(\mathbf{0}, \mathbf{I})] + \lambda_{\mathcal{Z}}\, d_{\cos}(\mathbf{z}, \boldsymbol{\phi}(\boldsymbol{\theta}(\mathbf{z}))),
\end{aligned} \tag{2}$$

where $\mathbf{x}$ is sampled from the training set, $\hat{\mathbf{x}} \sim q_\theta(\mathbf{x} \,|\, \mathbf{z})$, $\mathbf{z} \sim \mathrm{N}(\mathbf{0}, \mathbf{I})$, $d_{\mathcal{X}}$ is L1 or L2 distance, and $d_{\cos}$ is cosine distance. Since the model allows sampling from latent space (similarly to VAEs and GANs), the KL divergence terms during training can be calculated from empirical distributions, despite the model itself being fully deterministic. Yet we retain, in principle, the full information about the image, at every stage of the processing. For an image encoded into a latent vector $\mathbf{z}$, decoded back to image space as $\hat{\mathbf{x}}$, and re-encoded as latent vector $\mathbf{z}'$, it is possible and desirable to require that $\mathbf{z}$ is as close to $\mathbf{z}'$ as possible, yielding the latent reconstruction error $d_{\cos}(\mathbf{z}, \boldsymbol{\phi}(\boldsymbol{\theta}(\mathbf{z})))$.

We now show how to encourage the latent space to become hierarchically disentangled with respect to the levels of image detail, allowing one to separately retrieve 'coarse' and 'fine' aspects of a latent code. This allows for *e.g.* conditional sampling by fixing the latent code for specific layers of the decoder, or mixing the scale-specific features of two or more input images—impossible feats for a traditional autoencoder with mutually entangled decoder layers.

The latent reconstruction error as such can be trivially generalized to any layer of $\boldsymbol{\theta}$. We simply pick the layer of measurement, and from there, pass the sample once through a full encoder-decoder cycle. But now, in the automodulator, latent codes can be introduced on a per-layer basis, enabling more powerful reconstruction measurements. (Without loss of generality, here we only consider mixtures of 2 codes.) We can present a decoder (Fig. 1b) with N layers, split after the j^{th} one, as a composition of $\boldsymbol{\theta}_{j+1:N}(\boldsymbol{\theta}_{1:j}(\boldsymbol{\xi}^{(0)}, \mathbf{z}_A), \mathbf{z}_B)$. Crucially, we can choose $\mathbf{z}_A \neq \mathbf{z}_B$ (extending the method of [10]), such as $\mathbf{z}_A = \boldsymbol{\phi}(\mathbf{x}_A)$ and $\mathbf{z}_B = \boldsymbol{\phi}(\mathbf{x}_B)$ for (image) inputs $\mathbf{x}_A \neq \mathbf{x}_B$. Because the earlier layers #1:j operate on image content at lower resolutions, the output fusion image mixes the 'coarse' features of $\mathbf{z}_A$ with 'fine' features of $\mathbf{z}_B$. Now, $\mathbf{z}$ holds feature information at different levels of detail, some of which are mutually independent. Hence, when re-encoding an image, we should



(a) Architecture (b) 1-pass flow (c) 2-pass flow

Figure 1: (a) The autoencoder-like usage of the model. (b) Modulations in the decoder can come from different latent vectors. This can be leveraged in feature/style mixing, conditional sampling, and during the model training (first pass). (c) The second pass during training.

keep the representation of those levels disentangled in $\mathbf{z}$, even if they originate from separate source images. Hence, when we re-encode the fusion image into $\hat{\mathbf{z}}_{AB}$, and decode once more, the output of $\boldsymbol{\theta}_{1:j}(\boldsymbol{\xi}^{(0)}, \hat{\mathbf{z}}_{AB})$ should be unaffected by $\mathbf{z}_B$.

This leads to the following conjecture. Assume that the described network reconstructs input samples perfectly, *i.e.* $\mathbf{x} = \boldsymbol{\theta}(\boldsymbol{\phi}(\mathbf{x}))$. Any $\mathbf{z}_A$ and $\mathbf{z}_B$ can be mixed, decoded and re-encoded as

$$\hat{\mathbf{z}}_{AB} \sim q_\phi(\mathbf{z} \,|\, \mathbf{x})\, q_{\theta_{j+1:N}}(\mathbf{x} \,|\, \boldsymbol{\xi}^{(j)}, \mathbf{z}_B)\, q_{\theta_{1:j}}(\boldsymbol{\xi}^{(j)} \,|\, \boldsymbol{\xi}^{(0)}, \mathbf{z}_A). \tag{3}$$

Now, between $\boldsymbol{\xi}^{(j)}$ of the first and $\boldsymbol{\xi}^{(j)}$ of the second pass (see Fig. 1c), the mutual information is

$$I[q_{\theta_{1:j}}(\boldsymbol{\xi}^{(j)} \,|\, \boldsymbol{\xi}^{(0)}, \hat{\mathbf{z}}_{AB}); q_{\theta_{1:j}}(\boldsymbol{\xi}^{(j)} \,|\, \boldsymbol{\xi}^{(0)}, \mathbf{z}_A)]. \tag{4}$$

With $\mathbf{z}_A, \mathbf{z}_B \sim \mathrm{N}(\mathbf{0}, \mathbf{I})$, for each j, we can now maximize (4) by minimizing

$$\mathcal{L}_j = d(\boldsymbol{\theta}_{1:j}(\boldsymbol{\xi}^{(0)}, \hat{\mathbf{z}}_{AB}), \boldsymbol{\theta}_{1:j}(\boldsymbol{\xi}^{(0)}, \mathbf{z}_A)) \tag{5}$$

for some distance function d (here, L2 norm). In other words, the fusion image can be encoded into a new latent vector $\hat{\mathbf{z}}_{AB}$ in such a way that, at each layer, the decoder will treat the new code similarly to whichever of the original two separate latent codes was originally used there (see Fig. 1b). For a perfect network, $\mathcal{L}_j$ can be viewed as a layer entanglement error. Randomizing j during the training, we can measure $\mathcal{L}_j$ for any layers of the decoder.

Final Unsupervised Automodulator Training Loss We chose a potentially more robust image reconstruction loss d_ρ of [1] instead of L1/L2. d_ρ generalizes various norms and exposes robustness as an explicit continuous parameter vector $\boldsymbol{\alpha}$. The complete loss functions are

$$\begin{aligned}\mathcal{L}_\phi &= \max(-M_{\text{gap}}, \mathrm{D}_{\text{KL}}[q_\phi(\mathbf{z} \,|\, \mathbf{x}) \,\|\, \mathrm{N}(\mathbf{0}, \mathbf{I})] - \mathrm{D}_{\text{KL}}[q_\phi(\mathbf{z} \,|\, \hat{\mathbf{x}}) \,\|\, \mathrm{N}(\mathbf{0}, \mathbf{I})]) + \lambda_{\mathcal{X}}\, d_\rho(\mathbf{x}, \boldsymbol{\theta}(\boldsymbol{\phi}(\mathbf{x}))) \\ \mathcal{L}_\theta &= \mathrm{D}_{\text{KL}}[q_\phi(\mathbf{z} \,|\, \hat{\mathbf{x}}) \,\|\, \mathrm{N}(\mathbf{0}, \mathbf{I})] + \lambda_{\mathcal{Z}}\, d_{\cos}(\mathbf{z}, \boldsymbol{\phi}(\boldsymbol{\theta}(\mathbf{z}))) + \mathcal{L}_j,\end{aligned} \tag{6}$$

where $\hat{\mathbf{x}}_{1:\frac{3}{4}M} \sim q_\theta(\mathbf{x} \,|\, \mathbf{z})$ with $\mathbf{z} \sim \mathrm{N}(\mathbf{0}, \mathbf{I})$, and $\hat{\mathbf{x}}_{\frac{3}{4}M:M} \sim q_\theta(\mathbf{x} \,|\, \hat{\mathbf{z}}_{AB})$, with a set 3:4 ratio of regular and mixed samples of batch size M and $j \sim \mathrm{U}\{1, N\}$. (A smaller ratio appeared to disproportionately slow the overall learning, while a higher ratio could result in a very small number of mixed samples in high resolutions.) Margin $M_{\text{gap}} = 0.5$ in low resolutions, then 0.2 for 128^2 and 0.4 for 256^2 (see [5]). To avoid discontinuities in $\boldsymbol{\alpha}$, we utilize a progressively-growing variation of d_ρ, where we first learn the $\boldsymbol{\alpha}$ in the lower resolutions (*e.g.*, 4×4). Each α_i corresponds to one pixel. Then, when switching to the higher resolution stage, we take take each parameter α_i that corresponds to pixels $p_{x,y}$ in the lower resolution, to initialize the $\alpha_j^{1\times 4}$ that, in the higher resolution, corresponds to $p_{x,y}$, $p_{x+1,y}$, $p_{x,y+1}$ and $p_{x+1,y+1}$, respectively.

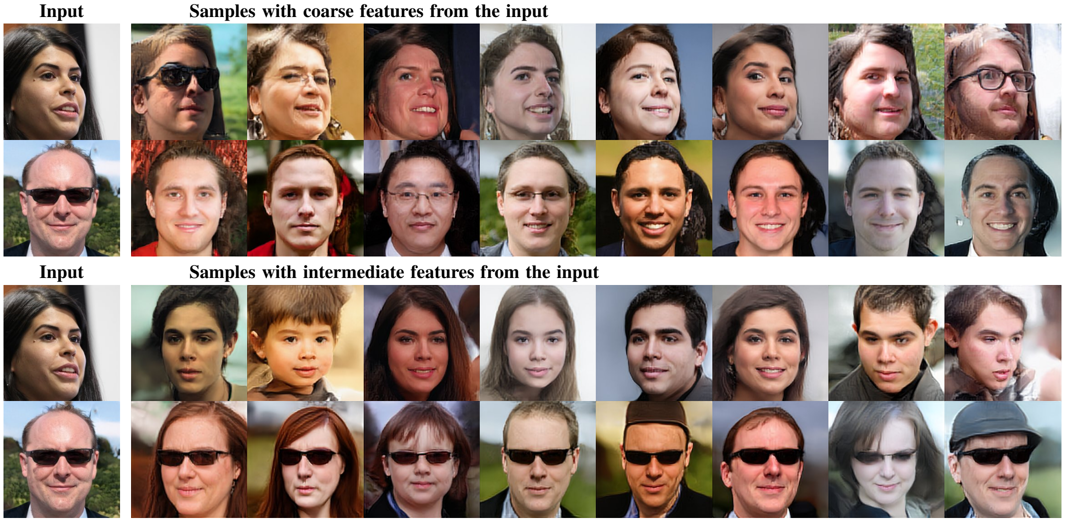

Input
Samples with coarse features from the input
Input
Samples with intermediate features from the input

Figure 2: Conditional sampling of 256×256 random face images based on 'coarse' (latent resolutions $4{\times}4$ – $8{\times}8$) and 'intermediate' ($16{\times}16$ – $32{\times}32$) latent features of the fixed input. The input image controls the coarse features (such as head shape, pose, gender) on the top and more fine features (expressions, accessories, eyebrows) on the bottom.

3 Experiments

We trained our model on the FFHQ faces data set [10] ($256{\times}256$ resolution) and CELEBA-HQ ($128{\times}128$). For both, we confirmed that it learns to produce sharp reconstructions and random samples. We then confirmed that the model can be used effectively for style mixing of images (see the Supplement), and finally evaluated the model performance for the task of image sampling conditioned on a certain level of features of an input face image. We feed the input face to the model at layers #1:2 (Fig. 2 top) on one run and at layers #3:4 (Fig. 2 bottom) on another. The latent for the other layers comes from random $\mathbf{z} \sim \mathrm{N}(\mathbf{0}, \mathbf{I})$. Here, we show two separate face identities, on the two levels of detail (Fig. 2).

Table 1: Comparison of random sampling performance via Fréchet Inception Distance (FID) and latent space structure via Perceptual path length (PPL) for images on CELEBA-HQ and LSUN Bedrooms between StyleGAN [10], Balanced PIONEER, PGGAN [9], GLOW [12] and our method. The resolutions are 128^2 for PPL and 256^2 for baseline FIDs [5]. FID for our method is based on 128^2. Autoencoders compared separately. 50k samples used for FID, 100k for PPL. For all metrics, **smaller is better**.

	FID (CELEBA-HQ)	FID (LSUN)	PPL (CELEBA-HQ)
StyleGAN	**5.06**	**2.65**	**50.08**
PGGAN	8.03	8.34	81.33
GLOW	68.93	—	138.21
Balanced PIONEER	**25.25**	**17.89**	92.84
Automodulator (ours)	28.84	30.13	**62.76**

Additionally, in CELEBA-HQ and LSUN Bedrooms [14], we quantify the image quality and diversity of random draws from our model with the Fréchet inception distance (FID, [6]), and for CELEBA-HQ, also the degree of disentanglement with Perceptual Path Length (PPL, [10]). Our method clearly outperforms the baselines (except for StyleGAN) in terms of PPL due to the latent space disentanglement, although the more advanced network architecture comes at the cost of worse FID results than the baseline PIONEER [5] (Table 1), even in the lower resolution. However, given that also in StyleGAN [10], the comparable architecture improvements made FID only slightly better, and style-mixing during training actually made FID worse, we believe the regular FID is not ideal for measuring models with style-mixing capabilities.

4 Discussion and Conclusion

We have introduced a novel method that uses architectural advancements for a new way of enforcing disentanglement of decoder layers of an autoencoder. In our model, latent variables affect the output via the 1st and 2nd moment of the intermediate decoder layers, allowing novel constraints for the intermediate outputs. Since our method takes additional advantage of the full encoder-decoder loop that the regular GAN architectures lack, it is theoretically stronger than the related ‘style-mixing loss’ of [10]. We applied the model to the task of conditional face image generation with convincing results. We believe that the range of applications of our model is far wider than this first application, making this family of autoencoders a viable alternative to state-of-the-art autoencoders and GANs.

Acknowledgments

The authors wish to acknowledge Aalto Science-IT project and CSC – IT Center for Science, Finland, for computational resources. Authors acknowledge funding from GenMind Ltd. This research was supported by the Academy of Finland grants 308640, 324345, 277685, and 295081.

References

[1] J. T. Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4331–4339, 2019.

[2] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2672–2680, 2014.

[3] A. Heljakka, A. Solin, and J. Kannala. Pioneer networks: Progressively growing generative autoencoder. In *Asian Conference on Computer Vision (ACCV)*, pages 22–38, 2018.

[4] A. Heljakka, A. Solin, and J. Kannala. Recursive chaining of reversible image-to-image translators for face aging. In *Proceedings of Advanced Concepts for Intelligent Vision Systems*, pages 309–320, 2018.

[5] A. Heljakka, A. Solin, and J. Kannala. Towards photographic image manipulation with balanced growing of generative autoencoders. *arXiv preprint arXiv:1904.06145*, 2019.

[6] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6626–6637, 2017.

[7] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1501–1510, 2017.

[8] D. Jimenez Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning (ICML)*, pages 1278–1286, 2014.

[9] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*, 2018.

[10] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4401–4410, 2019.

[11] D. Kingma and M. Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

[12] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 10236–10245. 2018.

[13] D. Ulyanov, A. Vedaldi, and V. Lempitsky. It takes (only) two: Adversarial generator-encoder networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, pages 1250–1257, 2018.

[14] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Supplementary Material for Conditional Image Sampling by Deep Automodulators

A Random samples

In addition to the conditional sampling described in Sec. 3, our model is also naturally capable of unconditional random sampling from regular unit Gaussian distribution (Fig. 3). The samples here indicate the full range of samples and face features the model can support. This should be contrasted with the narrow range of the conditional case.

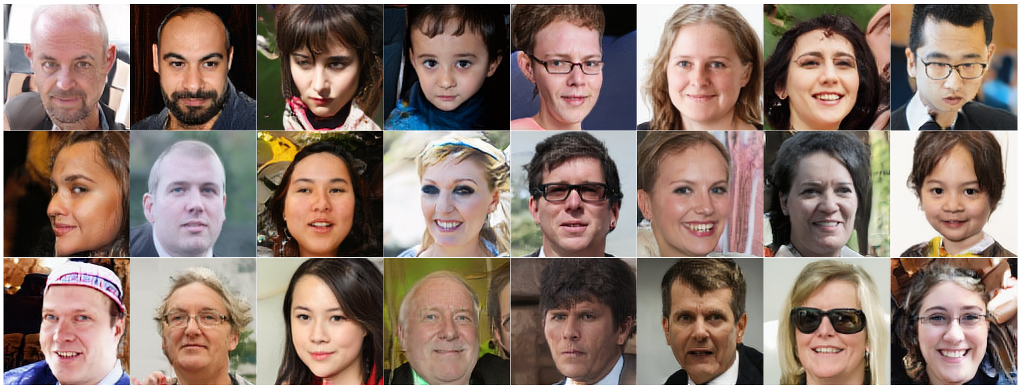


Figure 3: Random uncurated FFHQ samples at 256×256.

B Style mixing and interpolation

We can follow the methodology similar to the conditional sampling, and mix specific input faces so that the coarse, intermediate or fine layers of decoder use one input, and the rest of the layers use the other (Fig. 4). For comparison, we use the same input images as [10]. Importantly, the model in [10] cannot take real inputs, so the mix is actually done between images created by the model itself. For our model, those images appear as completely new test images.

We also show regular latent space interpolations between the reconstructions of new input images (Fig. 5).

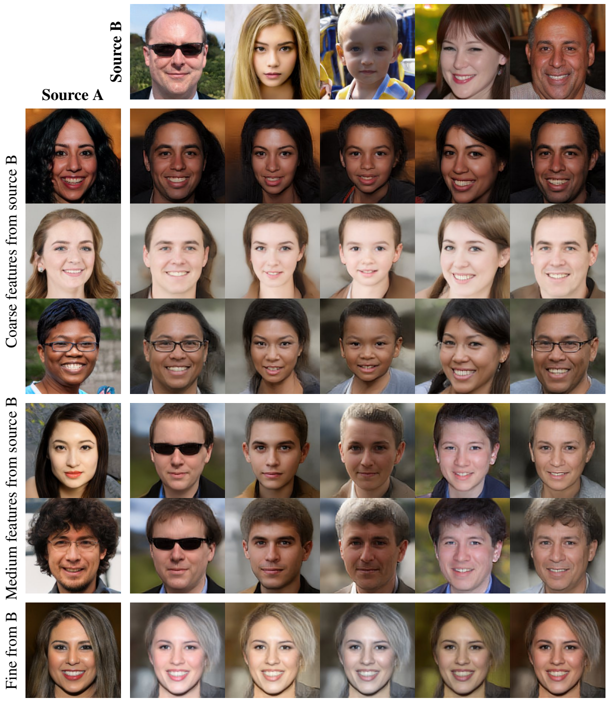


Figure 4: Style mix example, using the same input images as in [10]. Note that in [10], the source images were in fact generated by the network itself, and then mixed by the same network. For our model, these are novel input images, and the style mix problem is hence fundamentally harder.

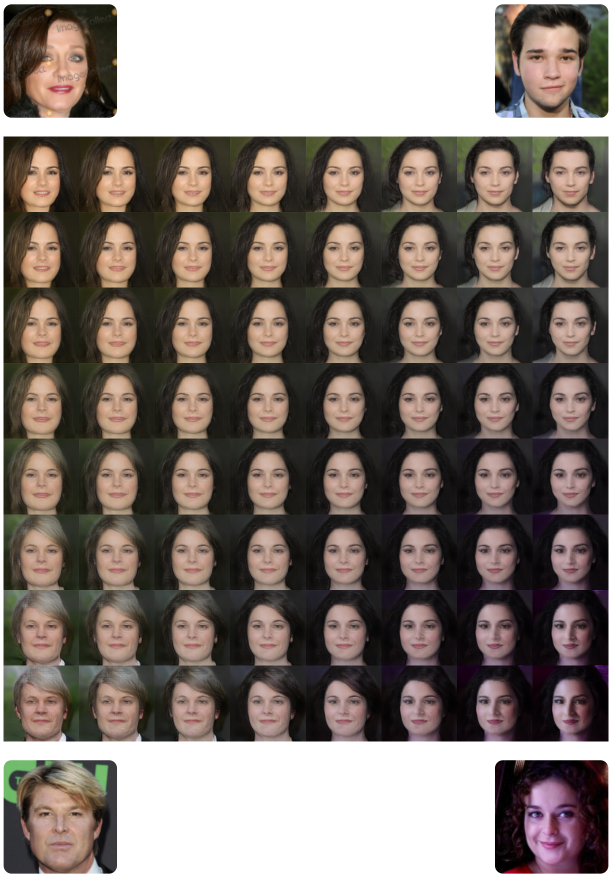


Figure 5: Interpolation between random test set CelebA-HQ images in 128×128 (in the corners) which the model has not seen during training. The model captures most of the salient features in the reconstructions and produces smooth interpolations at all points in the traversed space. The images in the corners of the square grid are the reconstructions of the adjacent input image, while the other images are interpolated between those.