Heljakka, Ari; Solin, Arno; Kannala, Juho

Pioneer Networks

Published in:
Computer Vision – ACCV 2018 - 14th Asian Conference on Computer Vision, Revised Selected Papers

DOI:
10.1007/978-3-030-20887-5_2

Published: 01/01/2019

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Heljakka, A., Solin, A., & Kannala, J. (2019). Pioneer Networks: Progressively Growing Generative Autoencoder. In G. Mori, C. V. Jawahar, K. Schindler, & H. Li (Eds.), Computer Vision – ACCV 2018 - 14th Asian Conference on Computer Vision, Revised Selected Papers (pp. 22-38). (Lecture notes in computer science; Vol. 11361). Springer. https://doi.org/10.1007/978-3-030-20887-5_2

Pioneer Networks: Progressively Growing Generative Autoencoder

Ari Heljakka[1,2], Arno Solin[1], and Juho Kannala[1]

[1] Department of Computer Science, Aalto University, Espoo, Finland
{ari.heljakka, arno.solin, juho.kannala}@aalto.fi
[2] GenMind Ltd

Abstract. We introduce a novel generative autoencoder network model that learns to encode and reconstruct images with high quality and resolution, and supports smooth random sampling from the latent space of the encoder. Generative adversarial networks (GANs) are known for their ability to simulate random high-quality images, but they cannot reconstruct existing images. Previous works have attempted to extend GANs to support such inference but, so far, have not delivered satisfactory high-quality results. Instead, we propose the Progressively Growing Generative Autoencoder (PIONEER) network which achieves high-quality reconstruction with 128×128 images without requiring a GAN discriminator. We merge recent techniques for progressively building up the parts of the network with the recently introduced adversarial encoder–generator network. The ability to reconstruct input images is crucial in many real-world applications, and allows for precise intelligent manipulation of existing images. We show promising results in image synthesis and inference, with state-of-the-art results in CELEBA inference tasks.

Keywords: Computer vision · Autoencoder · Generative models.

1 Introduction

Recent progress in generative image modelling and synthesis using generative adversarial networks (GANs, [7]) has taken us closer to robust high-quality image generation. In particular, progressively growing GANs (ProgGAN, [11]) can synthesize realistic high-resolution images with unprecedented quality. For example, given a training dataset of real face images, the models learnt by ProgGAN are capable of synthesizing face images that are visually indistinguishable from face images of real people.

However, GANs have no inference capability. While useful for understanding representations and generating content for training other models, the capability for realistic image synthesis alone is not sufficient for most applications. Indeed, in most computer vision tasks, the learnt models are used for feature extraction from existing real images. This motivates generative autoencoder models that allow both generation and reconstruction so that the mapping between the latent feature space and image space is bi-directional. For example, image enhancement and

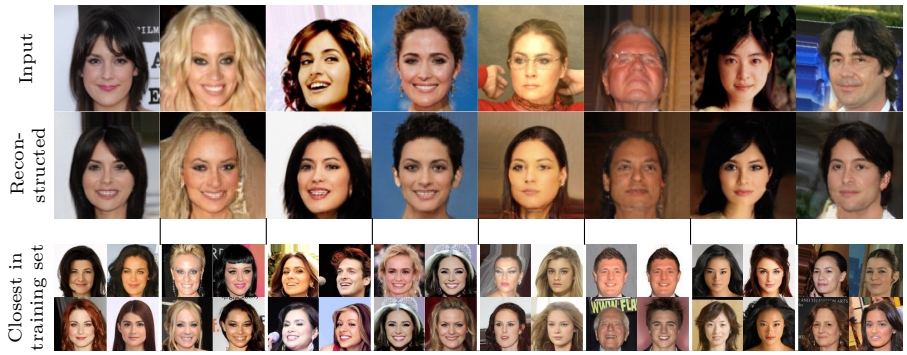


Fig. 1. Examples of PIONEER network reconstruction quality in 128×128 resolution (randomly chosen images from the CELEBA test set). Here, images are encoded into 512-dimensional latent feature vector and simply decoded back to the original dimensionality. Below each image pair, we show the four closest face matches to the input image in the training set (with respect to structural similarity [34] of the face, cropped as in [11]).

editing would benefit from generation and inference capabilities [3]. In addition, unsupervised learning of generative autoencoder models would be widely useful in semi-supervised recognition tasks. Yet, typically the models such as variational autoencoders (VAEs, [13,10]) generate samples not as realistic nor rich with fine details as those generated by GANs. Thus, there have been many efforts to combine GANs with autoencoder models [2,18,3,5,6,31,26], but none of them has reached results comparable to ProgGAN in quality.

In this paper, we propose the **P**rogress**I**vely gr**O**wi**N**g g**E**nerative auto**E**ncode**R** (PIONEER) network that extends the principle of progressive growing from purely generative GAN models to autoencoder models that allow both generation and inference. That is, we introduce a novel generative autoencoder network model that learns to encode and reconstruct images with high quality and resolution as well as to produce new high-quality random samples from the smooth latent space of the encoder. Our approach formulates its loss objective following [31], and we utilize spectral normalization [21] to stabilize training—to gain the same effect as the 'improved' Wasserstein loss [8] used in [11].

Similarly to [31], our approach contains only two networks, an encoder and a generator. The encoder learns a mapping from the image space to the latent space, while the generator learns the reciprocal mapping. Examples of reconstructions obtained by mapping a real input face image to the latent space and back using our learnt encoder and generator networks at 128×128 resolution are shown in Figure 1. Examples of synthetic face images generated from randomly sampled latent features by the generator are shown in Figure 3. In these examples, the model is trained using the CELEBA [16] and CELEBA-HQ [11] datasets in a completely unsupervised manner. We also demonstrate very smooth interpolation

between tuples of test images that the network has never seen before, a task that is difficult and tedious to carry out with GANs.

In summary, the key contributions and results of this paper are: *(i)* We propose a generative image autoencoder model whose architecture is built up progressively, with a balanced combination of reconstruction and adversarial losses, but without a separate GAN-like discriminator; *(ii)* We show that at least up to 128×128 resolution, this model can carry out inference on input images with sharp output, and up to 256×256 resolution, it can generate sharp images, while having a simpler architecture than the state-of-the-art of purely generative models; *(iii)* Our model gives improved image reconstruction results with larger image resolutions than previous state-of-the-art on CelebA. The PyTorch source code of our implementation is available at `https://aaltovision.github.io/pioneer`.

2 Related Work

Pioneer networks belong to the family of generative models, with variational autoencoders (VAEs), autoregressive models, GAN variants, and other GAN-like models (such as [14]). The core idea of a GAN is to jointly train so-called generator and discriminator networks so that the generator learns to output samples from the same distribution as the training set [7], when given random input vectors from a low-dimensional latent space, and the discriminator simultaneously learns to distinguish between the synthetic and real training samples. The generator and discriminator are differentiable, jointly learnt via backpropagation using alternating optimization of an adversarial loss, where the discriminator is updated to maximize the probability of correctly classifying real and synthetic samples and the generator is updated to maximize the probability of discriminator making a mistake. Upon convergence, the generator learns to produce samples that are indistinguishable from the training samples (within the limits of the discriminator network's capacity).

Making the aforementioned training process stable has been a challenge, but the Wasserstein GAN [1] improved the situation by adopting a smooth metric for the distance between the two probability distributions [8]. In Karras *et al.* [11], the Wasserstein GAN loss from [8] is combined with the idea of progressively growing the layers and image resolution of the generator and discriminator during training, yielding excellent image synthesis results. Progressive growing has been used successfully also, for example, by [33]. There is also a line of work on other regularizers that stabilize the training (*e.g.* [27,21,22]).

However, it is well understood that the capability for realistic image synthesis alone is not sufficient for applications and there is a need for better unsupervised feature learning methods that are able to capture the semantically relevant dependencies of input data into a compact latent representation [5]. In their basic form, GANs are not suitable for this purpose as they do not provide means of learning the inverse mapping that projects the data back to latent space.

Nevertheless, there have been many recent efforts which utilize adversarial training for learning bi-directional generative models that would allow both image

synthesis and reconstruction in a manner similar to autoencoders. For example, the recent works [5] and [6] simultaneously proposed an approach that employs three deep neural networks (generator, encoder, and discriminator) for learning bi-directional mappings between the data space and latent space. Instead of just samples, the discriminator is trained to discriminate tuples of samples with their latent codes, and it is shown that at the global optimum the generator and encoder learn to invert each other. Further, several others have proposed 3-network approaches that add some form of reconstruction loss and combine ideas of GAN and VAE: [2] extends VAE with a GAN-like discriminator for sample space (also used by [3]), [18,20] do the same with a GAN-like discriminator for the latent space, and [26] adds yet another discriminator (for the VAE likelihood term). While the previous methods have advanced the field, they still have not been able to simultaneously provide high quality results for both synthesis and reconstruction of high resolution images. Most of these methods struggle with even 64×64 images.

Recently, Ulyanov *et al.* [31] presented an autoencoder architecture that simply consists of two deep networks, a generator θ and encoder ϕ, representing mappings between the latent space and data space, and trained with a combination of adversarial optimization and reconstruction losses. That is, given the data distribution X and a simple prior distribution Z in the latent space, the updates for the generator aim to minimize the divergence between Z and $\boldsymbol{\phi}(\boldsymbol{\theta}(Z))$, whereas the updates for the encoder aim to minimize the divergence between Z and $\boldsymbol{\phi}(X)$ and simultaneously maximize the divergence between $\boldsymbol{\phi}(\boldsymbol{\theta}(Z))$ and $\boldsymbol{\phi}(X)$. In addition, the adversarial loss is supplemented with reconstruction losses both in the latent space and image space to ensure that the mappings are reciprocal (*i.e.* $\boldsymbol{\phi}(\boldsymbol{\theta}(\mathbf{z})) \simeq \mathbf{z}$ and $\boldsymbol{\theta}(\boldsymbol{\phi}(\mathbf{x})) \simeq \mathbf{x}$). The results of [31] are promising regarding both synthesis and reconstruction but the images still have low resolution. Scaling to higher resolutions requires a larger network which makes adversarial training less stable.

We combine the idea of progressive network growing [11] with the adversarial generator–encoder (AGE) networks of [31]. However, the combination is not straightforward, and we needed to identify a proper set of techniques to stabilize the training. In summary, our contributions result in a model that is simpler than many previous ones (*e.g.* having a large discriminator network just for the purpose of training the generator is wasteful and can be avoided), provides better results than [31] already in small (64×64) resolutions, and enables training and good results with larger image resolutions than previously possible. The differences to [26], [5], and, for example, [29] are substantial enough to perceive by quick visual comparison.

3 Pioneer Networks

Our generative model achieves three key goals that define a good encoder–decoder model: *(i)* faithful reconstruction of the input sample, *(ii)* high sample quality (whether random samples or reconstructions), and *(iii)* rich representations. The

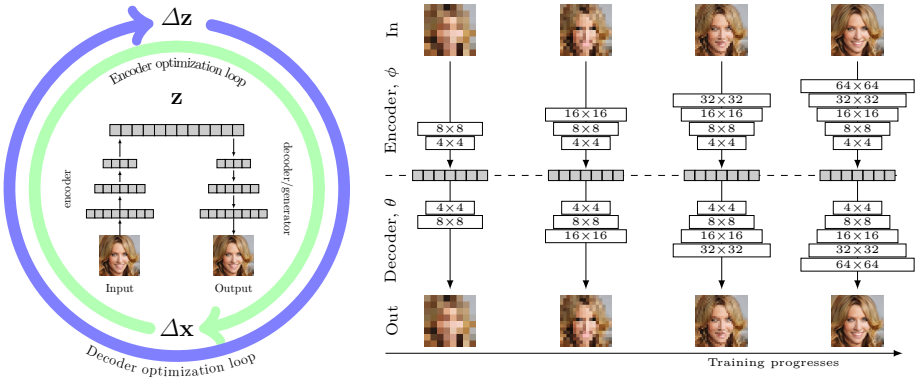


Fig. 2. The network grows in phases during which the image resolution doubles. The adversarial/reconstructive training criterion is continuously applied at each step, adapting to the present input–output resolution. The circular arrows illustrate the two modes of learning: *(i)* reconstruction of real training samples, and *(ii)* reconstruction of the latent representation of randomly generated samples.

final item can be reformulated as a 'well-behaved' latent space that lends itself to high-quality interpolations between given test samples and captures the diversity of features present in the training set. Critically, these requirements are strictly parametrized by our target resolution—there are several models that achieve many of the said goals up to 32×32 image resolution, but very few that have shown good results beyond 64×64 resolution.

Pioneer networks achieve the reconstruction and representation goals up to 128×128 resolution and the random sample generation up to 256×256 resolution, while using a combination of simple principles. A conceptual description in the next subsection is followed by some theory (Sec. 3.2) and more practical implementation details (Sec. 3.3).

3.1 Intuition

The defining training and architecture principles of Pioneer networks are shown in Figure 2; on the left hand side, the competing objectives are presented in the double loop, and on the right, the progressively growing structure of the network is shown stepping up through $4\times4, 8\times8, 16\times16, \ldots$, doubling the resolution in each phase. The input $\mathbf{x}$ is squeezed through the encoder into a latent representation $\mathbf{z}$, which on the other hand is again decoded back to an image $\hat{\mathbf{x}}$. The motivation behind the progressively growing setup is to encourage the network to catch the fundamental structure and variation in the inputs at lower resolutions to help the additional layers specialize in fine-tuning and adding details and nuances when reaching the higher resolutions.

The network has encoder–decoder structure with no *ad hoc* components (such as separate discriminators as in [3,2,26,18,20]). Similar to GANs, the encoder

and decoder are not trained as one, but instead as if they were two competing networks. This requires the encoder to become sensitive to the difference between training samples and generated (decoded) samples, and the decoder to keep making the difference smaller and smaller. While GANs achieve this with the complexity cost of a separate discriminator network, we choose to just learn to encode the samples in a source-dependent manner. This encoding could be, then, followed by a classification layer, but instead we train the encoder so that the distribution of latent codes of training samples *approach* a certain reference distribution, while the distribution of codes of generated samples *diverges* from it (see AGE [31]).

3.2 Encoder–Decoder Losses

As in variational autoencoders, we choose the Kullback–Leibler (KL) divergence as the metric in latent space. Our reference distribution is unit Gaussian with a diagonal covariance matrix. Each sample $\mathbf{x} \in X$ is encoded into a latent vector $\mathbf{z} \in Z$, giving rise to the posterior distribution $q_\phi(\mathbf{z} \mid \mathbf{x})$ on a d-dimensional sphere. The KL-divergence between such a distribution and a d-dimensional unit Gaussian is (see the reasoning in [31], but with the following corrections):

$$\mathrm{KL}[q_\phi(\mathbf{z} \mid \mathbf{x}) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I})] = -\frac{d}{2} + \sum_{j=1}^{d} \left[\frac{\sigma_j^2 + \mu_j^2}{2} - \log(\sigma_j) \right], \tag{1}$$

where μ_j and σ_j are the empirical sample mean and standard deviation of the encoded samples in the latent vector space with respect to dimension $j = 1, 2, \ldots, d$, and $\mathcal{N}(\mathbf{0}, \mathbf{I})$ denotes the unit Gaussian.

The encoder ϕ and decoder θ are connected via two reconstruction error terms. We measure reconstruction error $L_\mathcal{X}$ with L1 distance in sample space $\mathcal{X}$ for the encoder, and code reconstruction error $L_\mathcal{Z}$ with cosine distance in latent code space $\mathcal{Z}$ for the decoder, as follows:

$$L_\mathcal{X}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{\mathbf{x} \sim X} \|\mathbf{x} - \boldsymbol{\theta}(\boldsymbol{\phi}(\mathbf{x}))\|_1, \tag{2}$$

$$L_\mathcal{Z}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{\mathbf{z} \sim Z}[1 - \mathbf{z}^\mathsf{T} \boldsymbol{\phi}(\boldsymbol{\theta}(\mathbf{z}))], \tag{3}$$

where X are the training samples and Z random latent vectors, with $\mathbf{z}$ and $\boldsymbol{\phi}(\mathbf{x})$ normalized to unity.

In other words, a training sample is encoded into the latent space and then decoded back into a generated sample. A random latent vector is decoded into a random generated sample that is then fed back to the encoder (Fig. 2). This provides an elegant solution to forcing the network to learn to reconstruct training images. The total loss function of the encoder L_ϕ and decoder L_θ are, then:

$$L_\phi = \quad \mathrm{KL}[q_\phi(\mathbf{z} \mid \mathbf{x}) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I})] - \mathrm{KL}[q_\phi(\mathbf{z} \mid \hat{\mathbf{x}}) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I})] + \lambda_\mathcal{X} L_\mathcal{X}, \tag{4}$$

$$L_\theta = -\mathrm{KL}[q_\phi(\mathbf{z} \mid \mathbf{x}) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I})] + \mathrm{KL}[q_\phi(\mathbf{z} \mid \hat{\mathbf{x}}) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I})] + \lambda_\mathcal{Z} L_\mathcal{Z}, \tag{5}$$

where $\mathbf{x} \sim X$ and $\hat{\mathbf{x}} = \boldsymbol{\theta}(\mathbf{z})$ with $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We fix the hyper-parameters $\lambda_{\mathcal{X}}$ and $\lambda_{\mathcal{Z}}$ so they can be read as scaling constants. In practical implementation, we can simplify the decoder loss to only account for

$$L_\theta = \mathrm{KL}[q_\phi(\mathbf{z} \mid \hat{\mathbf{x}}) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I})] + \lambda_{\mathcal{Z}} L_{\mathcal{Z}}. \tag{6}$$

The training is adversarial in the sense that we use each loss function in turn, first freezing the decoder weights and training only with the loss (4), and then freezing the encoder weights and training only with the loss (6).

However, in Ulyanov *et al.* [31], this approach was only shown to work with AGE on images up to 64×64 resolution. Beyond that, we need a larger network architecture, which is unlikely to work with AGE alone. We confirmed this by trying out a straightforward extension of AGE to 128×128 resolution (by visual examination and via results in Table 1). In contrast, to stabilize training, our model will increase the size of the network progressively, following [11], and utilize the following techniques.

3.3 Model and Training

The training uses a convolution–deconvolution architecture typically used in generative models, but here, the model is built up progressively during training, as in [11]. We start training on low resolution images (4×4), bypassing most of the network layers. We train each intermediate phase with the same number of samples. In the first half of each consequtive phase, we start by adding a trivial downsampling (encoder) and upsampling (decoder) layer, which we gradually replace by fading in the next convolutional–deconvolutional layers simultaneously in the encoder and the decoder, in lockstep with the input resolution which is also faded in gradually from the previous to the new doubled resolution (8×8 etc.). During the second half of each phase, the architecture remains unchanged. After the first half of the target resolution phase, we no longer change the architecture.

We train the encoder and the generator with loss (4) and (6) in turn, utilizing various stabilizing factors as follows. The architecture of the convolutional layers in Pioneer networks largely follows yet simplifies the symmetric structure in ProgGAN (see Table 2 of [11]), with the provision of replacing its discriminator with an encoder. This requires removing the binary classifier, allowing us to connect the encoder and decoder directly via the 512-dimensional latent vector. We also remove the minibatch standard deviation layer, as it is sensitive to batch-level statistics useful for a GAN discriminator but not for an encoder.

For stabilizing the training, we employ equalized learning rate and pixelwise feature vector normalization in the generator [11], buffer of images created by previous generators [28], and encoder spectral normalization [21]. We use ADAM [12] with $\beta_1 = 0, \beta_2 = 0.99, \epsilon = 10^{-8}$ and learning rate 0.001. We use 2 generator updates per 1 encoder update. For result visualization (but not training), we use an exponential running average for the weights of the generator over training steps as in [11]. Of these techniques, spectral normalization warrants some elaboration.

To stabilize the training of generative models, it is important to consider the function space within which the discriminator must fit in general, and, specifically,

Fig. 3. Randomly generated face image samples with PIONEER networks using CELEBA for training at resolutions 64×64 (top) and 128×128 (middle), and using CELEBA-HQ for 256×256 (bottom).

controlling its Lipschitz constant. ProgGAN uses improved Wasserstein loss [8] to keep the Lipschitz constant close to unity. However, this loss formulation is not immediately applicable to the slightly more complex AGE-style loss formulation, so instead, we adopted GAN spectral normalization [21] to serve the same purpose. In this spectral normalization approach, the spectral norm of each layer of the encoder (discriminator) network is constrained directly at each computation pass, allowing the network to keep the Lipschitz constant under control. Crucially, spectral normalization does not regularize the network via learnable parameters, but affects the scaling of network weights in a data-dependent manner.

In our experiments, it was evident that without such a stabilizing factor, the progressive training would not remain stable beyond 64×64 resolution. Spectral normalization solved this problem unambiguously: without it, the training of the network was consistently failing, while with it, the training almost consistently converged. Other strong stabilization methods, such as the penalty on the weighted gradient norm [27], might have worked here as well.

4 Experiments

PIONEER networks are more most immediately applicable to learning image datasets with non-trivial resolutions, such as CELEBA [16], LSUN [32], and IMAGENET. Here, we run experiments on CELEBA and CELEBA-HQ [11] (with training/testing split 27000/3000) and LSUN bedrooms. For comparing with previous works, we also include CIFAR-10, although its low-resolution images (32×32) were not expected to be most relevant for the present work.

Training with high resolutions is relatively slow in both ProgGAN and our method, but we believe that significant speed optimization is possible in future

work. In fact, it is noteworthy that you *can* train these models for a long time without running into typical GAN problems, such as 'mode collapse' or ending up oscillating around a clearly suboptimal point. We trained the PIONEER model on CELEBA with one Titan V GPU for 5 days up to 64×64 resolution (172 epochs), and another 8 days for 128×128 resolution. We separately trained on CELEBA-HQ up to 256×256 resolution with four Tesla P100 GPUs for 10 days (1600 epochs), and on LSUN with two Tesla P100 GPUs for 9 days.

Throughout the training, we kept the hyper-parameters fixed at $\lambda_{\mathcal{Z}} = 1000\,d$ and $\lambda_{\mathcal{X}} = 10\,d$, where d is the dimensionality of the latent space (512), taking advantage of the hyper-parameter search done by [31]. After the progressive growth phase of the training, we switched to $\lambda_{\mathcal{X}} = 15\,d$ to emphasize sample reconstruction [30].

4.1 CelebA and CelebA-HQ

The CELEBA dataset [16] contains over 200k images with various resolutions that can be square-cropped to 128×128. CELEBA-HQ [11] is a subset of 30k of those images that have been improved and upscaled to 1024×1024 resolution. We train with CELEBA up to 128×128 resolution, and with CELEBA-HQ up to 256×256. In order to compare with previous works, we also trained our network for 64×64 images from CELEBA.

We ran our experiments as follows. Following the approach described in Section 3.3, we trained the network progressively through each intermediate resolution until we reach the target resolution (64×64, 128×128, or 256×256), for the same number of steps in each stage. For the final stage with the target resolution, we would continue training for as long as the Fréchet Inception Distance (FID, [9]) measures of the randomly generated samples showed improvements. During the progression of the input resolution, we adapted minibatch size to accommodate for the available memory.

For random sampling metrics, we use FID and Sliced Wasserstein Distance (SWD, [23]) between the training distribution and the generated distribution. FID measures the sample quality and diversity, while SWD measures the similarity in terms of Wasserstein distance (earth mover's distance). Batch size is 10000 for FID and 16384 for SWD. For reconstruction metrics, we use the root-mean-square error (RMSE) between the original and the reconstructed image.

We present our results in three ways. First, the model must be able to reconstruct random test set images and retain both sufficient quality and faithfulness of the reconstruction. Often, there is a trade-off between the two [25]. Previous models have often seemed to excel with respect to the quality of the reconstruction image, but in fact, the reconstruction turns out to be very different from the original (such as a different person's face). Second, we must be able to randomly sample images from the latent space of the model, and achieve sufficient quality and diversity of the images. Third, due to its inference capability, PIONEER networks can show interpolated results between input images without any additional tricks, such as first solving a regression optimization problem for the image, as often done with GANs (*e.g.* [24]).

Table 1. Comparison of Fréchet Inception Distance (FID) against 10,000 training samples, Sliced Wasserstein Distance (SWD) against 16384 samples, and root-mean-square error (RMSE) on test set, in the 64×64 and 128×128 CelebA dataset on inference-capable networks. ProgGAN with L1 regression has the best overall sample quality (in FID/SWD), but not best reconstruction capability (in RMSE). A pretrained model by the author of [31] was used for AGE for 64×64. For 128×128, we enlarged the AGE network to account for the larger inputs and a 512-dimensional latent vector, trained until the training became unstable. ALI was trained on default CelebA settings following [6] for 123 epochs. The error indicates one standard deviation for separate sampling batches of a single (best) trained model. For all numbers, **smaller is better**.

	64×64			128×128		
	FID	SWD	RMSE	FID	SWD	RMSE
ALI	58.88 ± 0.19	25.58 ± 0.35	18.00 ± 0.21	—	—	—
AGE	26.53 ± 0.08	17.87 ± 0.11	4.97 ± 0.06	154.79 ± 0.43	22.33 ± 0.74	9.09 ± 0.07
ProgGAN/L1	$\mathbf{7.98 \pm 0.06}$	$\mathbf{3.54 \pm 0.40}$	2.78 ± 0.05	—	—	—
Pioneer	8.09 ± 0.05	5.18 ± 0.19	$\mathbf{1.82 \pm 0.02}$	$\mathbf{23.15 \pm 0.15}$	$\mathbf{10.99 \pm 0.44}$	$\mathbf{8.24 \pm 0.15}$

Reconstruction. Given an unseen test image, the model should be able to encode the relevant information (such as hair color, facial expression, *etc.*) and decode it into a natural-looking face image expressing the features. Unlike in image compression, the model does not aim to replicate the input image *per se*, but capture the essentials. In Figure 1, we show Pioneer reconstructions for CelebA 128×128 test images, coupled with the four closest samples in the training set (in terms of structural similarity [34] of the face as cropped in [11]).

We compare reconstructions against inference-capable models: AGE [31] and ALI [6]. We also train ProgGAN for reconstruction as follows (compare to *e.g.*, [24,15,17,4]). We train the network normally until convergence, and then use the latent vector of the discriminator also as the latent input for the generator (properly normalized). Finally, we re-train the discriminator–generator network as an autoencoder that attempts to reconstruct input images, with L1 reconstruction loss. When re-training, we only modify the discriminator subnetwork, since allowing the generator to change would inevitably lead to lower-quality generated images. (We also tried training a fully connected layer on top of the existing hidden layer, but training became almost prohibitively slow without improved results.) Like most of the previous results, we find that the network (ProgGAN/L1) can fairly well reconstruct samples that it has generated itself, but performs much worse when given new real input images.

For networks that support both inference and generation, we can feed input images and evaluate the output image. In Figure 4, we show the output of each network for the given random CelebA test set images. As seen from the figure, at 64×64 resolution, Pioneer outperforms the baseline networks in terms of the combined output quality and faithfulness of the reconstruction. At 64×64 resolution, Pioneer's FID score of 8.09 in Table 1 outperforms AGE and ALI, the relevant inference baselines. ProgGAN/L1 outperforms the rest in sample quality (FID/SWD), but is worse in faithfulness (RMSE).

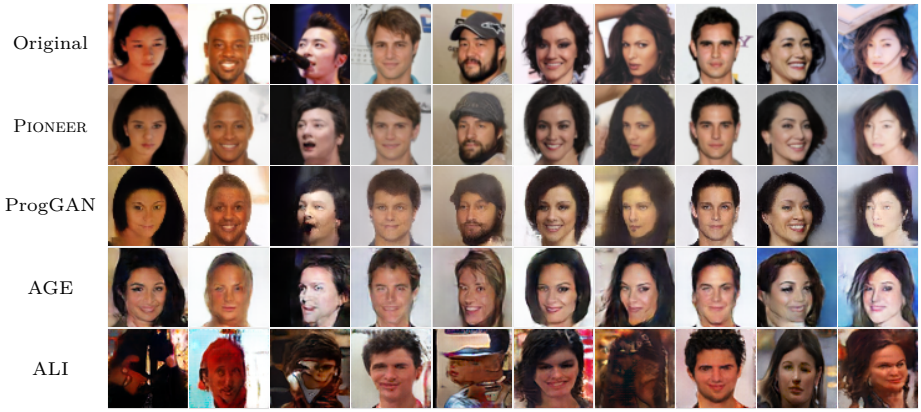
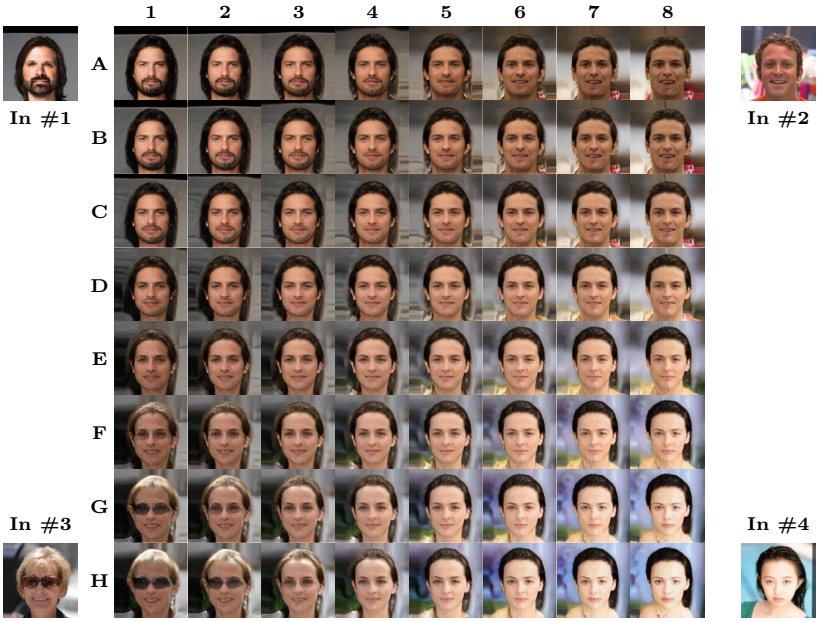


Fig. 4. Comparison of reconstruction quality between Pioneer, ALI, and AGE in 64×64. The first row in each set shows examples from the test set of CelebA (not cherry-picked). The reproduced images of Pioneer are much closer to the original than those of AGE or ALI. Note the differences in handling of the 5th image from the left. ALI and AGE were trained as in Table 1. (For more examples, see Supplementary)

Without modifications, ALI and AGE have not thus far been shown to work with 128×128 resolution. We managed to run AGE for 128×128 resolution by enlarging the network to account for the larger inputs and a 512-dimensional latent vector, and trained until the training became unstable. For ALI, enlarging the network for 128×128 was not tried. ProgGAN excels in sample generation for higher resolutions, but as discussed, it is not designed for reconstruction or inference. Therefore we ran it only for 64×64, already showing this difference.

Dreaming up random samples. A model that focuses on reconstruction is unlikely to match the quality of the models that only focus on random sample generation. Even though our focus is on excelling in the former category, we do not fall far behind the state-of-the-art provided by ProgGAN in generating new samples. Figure 3 shows samples generated by Pioneer at 64×64, 128×128, and 256×256 resolutions. The ProgGAN SWD results in [11] were based on a more aggressive cropping of the dataset, so the values are not comparable. For AGE and ALI, the FID and SWD scores are clearly worse (see Table 1) even at low resolutions, and the methods do not generalize well to higher resolutions.

Inference capabilities. Finally, we provide an example of input-based interpolation between different (unseen) test images. In Figure 5 we have four different test images, one in each corner of the tile figure. Thus image A1 corresponds to the reconstruction of Input #1, A8 to Input #2, H1 to Input #3, and H8 to Input #4. The rest of the images are produced by interpolating between the reconstructions in the latent space—for example, between A1 and A8. As can be seen in the figure, the latent space is well-behaved and even the glasses in



(Best viewed in high resolution / zoomed-in.)

Fig. 5. Interpolation study on test set input images at 128×128 resolution. Unlike many works, we interpolate between the (reconstructions of) unseen test images given as input—not between images the network has generated on its own.

Input #3 do not cause problems. We emphasize that compared to many GAN methods, the interpolations in PIONEER can be done elegantly and without separate optimization stage needed for each input sample.

4.2 LSUN Bedrooms

The LSUN dataset [32] contains images of various categories in 256×256 resolution or higher. We choose the category of bedrooms, often used for testing generative models. For humans, comparing randomly generated samples is more difficult on this dataset than with faces, so quantitative metrics are important to separate between the subtle differences in quality and diversity of captured features.

We ran the LSUN training similarly to CELEBA, but with only a single target resolution of 128×128. We present randomly generated samples from LSUN bedrooms (Fig. 6) at 128×128 resolution. Comparing to the non-progressive GANs of [8] and [19], we see that PIONEER output quality visually matches them, while falling slightly behind the fully generative ProgGAN, as expected. The FID of 37.50 was reached with no hyper-parameter tuning specific to LSUN.

For networks that support both inference and generation, we would not expect to achieve the same quality metrics as with purely generative models, so these results are not directly comparable.

Fig. 6. Generated images of LSUN bedrooms at 128×128 resolution. Results can be compared to the image arrays in [19], [8], and [11].

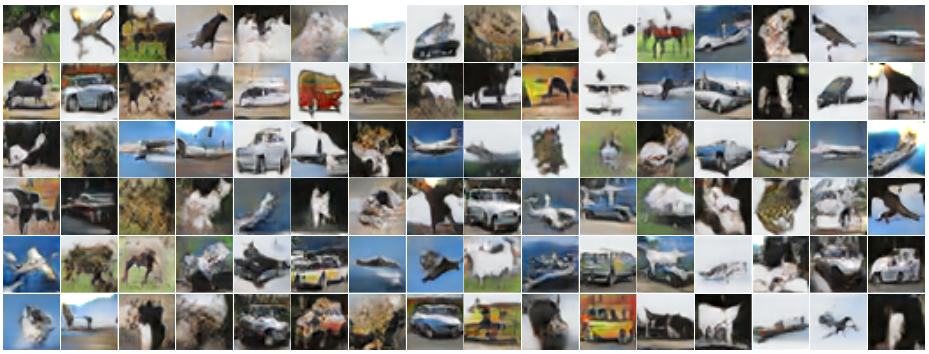


Fig. 7. Generated images of CIFAR-10 at 32×32 resolution.

4.3 Cifar-10

The CIFAR-10 dataset contains 60,000 labeled images at 32×32 resolution, spanning 10 classes. As our method is fully supervised, we do not utilize the label information. During the training, we found that progressive growing seemed to provide no benefits. Therefore, we trained the PIONEER model otherwise as normal, but started at 32×32 resolution and did not use progressive growing.

We used the same architecture, losses and algorithm as for the other datasets, instead of trying to optimize our approach to get the best results in CIFAR-10. We confirmed that the approach works, but it is not particularly suitable for this kind of a dataset without further modifications. Generated samples are provided in Figure 7. We believe that with some natural modifications, the model will be able to compete with GAN-based methods, but we leave this for our future work.

5 Discussion and Conclusion

In this paper, we proposed a generative image autoencoder model that is trained with the combination of adversarial and reconstruction losses in sample and latent space, using progressive growing of generator and encoder networks, but without a separate GAN-like discriminator. We showed that this model can both generate sharp images—at least up to 256×256 resolution—and carry out inference on input images at least up to 128×128 resolution with sharp output, while having a simpler architecture than the state-of-the-art of purely generative models [11]. We demonstrated the inference via sample reconstruction and smooth interpolation in the latent space, and showed the overall generative capability by generating new random samples from the latent space and measuring the quality and diversity of the generated distribution against baselines.

We emphasize that evaluation of generative models is heavily dependent on the resolution, and there is a multitude of models that have been shown to work on 64×64 resolution, but not on 128×128 or above. Reaching higher resolutions is not only a matter of raw compute, but the model needs to be able to cope with the increasing information and be regularised suitably in order not to loose the representative power or become instable.

We found that training is more stable using spectral normalization, which also suits our non-GAN loss architecture and loss. The model provides image reconstruction results with larger image resolutions than previous state-of-the-art. Importantly, our model has only few hyper-parameters and is robust to train. The only hyper-parameter that typically needs to be tuned between datasets is the number of epochs spent on intermediate resolutions. Our results indicate that the GAN paradigm of a separate discriminator network may not be necessary for learning to infer and generate image data sets. GANs do currently remain the best option if one is only interested in generating random samples. Like GANs, our model is heavily based on the general idea of 'adversarial' training, construed as setting the generator–encoder pair up with opposite gradients to each other with respect to the source of the data (that is, simulated vs. observed).

As Karras *et al.* [11] point out for GANs, the principle of growing the network progressively may be more important than the specific loss function formulation. Likewise, even though the AGE formulation for the latent space loss metrics is relatively simple, we believe that there are many ways in which the encoder can be set up to achieve and exceed the results we have demonstrated here.

In future work, we will also continue training the network to carry out faithful reconstructions at 256×256, 512×512, and 1024×1024 resolutions, omitted from this paper primarily due to the extensive amount of computation (or preferably, further optimization) required. We will also further investigate whether the CelebA-HQ dataset is sufficiently diverse for this purpose.

Acknowledgments We thank Tero Karras, Dmitry Ulyanov, and Jaakko Lehtinen for fruitful discussions. We acknowledge the computational resources provided by the Aalto Science-IT project. Authors acknowledge funding from the Academy of Finland (grant numbers 308640 and 277685) and GenMind Ltd.

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International Conference on Machine Learning (ICML) (2017)
2. Boesen Lindbo Larsen, A., Kaae Sønderby, S., Larochelle, H., Winther, O.: Autoencoding beyond pixels using a learned similarity metric. In: International Conference on Machine Learning (ICML) (2016)
3. Brock, A., Lim, T., Ritchie, J.M., Weston, N.: Neural photo editing with introspective adversarial networks. In: International Conference on Learning Representations (ICLR) (2017)
4. Creswell, A., Bharath, A.A.: Inverting the generator of a generative adversarial network. In: NIPS 2016 Workshop on Adversarial Training (2016)
5. Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial feature learning. In: International Conference on Learning Representations (ICLR) (2017)
6. Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M., Courville, A.: Adversarially learned inference. In: International Conference on Learning Representations (ICLR) (2017)
7. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. In: Advances in Neural Information Processing Systems (NIPS) (2014)
8. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein GANs. In: Advances in Neural Information Processing Systems (NIPS) (2017)
9. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In: Advances in Neural Information Processing Systems (NIPS) (2017)
10. Jimenez Rezende, D., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. In: International Conference on Machine Learning (ICML) (2014)
11. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. In: International Conference on Learning Representations (ICLR) (2018)
12. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015)
13. Kingma, D., Welling, M.: Auto-encoding variational Bayes. In: International Conference on Learning Representations (ICLR) (2014), `https://arxiv.org/abs/1312.6114`
14. Li, Y., Swersky, K., Zemel, R.: Generative moment matching networks. In: International Conference on Machine Learning (ICML) (2015)
15. Lipton, Z.C., Tripathi, S.: Precise recovery of latent vectors from generative adversarial networks. In: International Conference on Learning Representations (ICLR) (2017)
16. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: International Conference on Computer Vision (ICCV) (2015)
17. Luo, J., Xu, Y., Tang, C., Lv, J.: Learning inverse mapping by autoencoder based generative adversarial nets. In: Neural Information Processing (ICONIP). Lecture Notes in Computer Science, vol. 10635 (2017)
18. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B.: Adversarial autoencoders. arXiv preprint arXiv:1511.05644 (2015)

19. Mao, X., Li, Q., Xie, H., Lau, R.Y.K., Wang, Z., Smolley, S.P.: Least squares generative adversarial networks. In: International Conference on Computer Vision (ICCV) (2017)
20. Mescheder, L., Nowozin, S., Geiger, A.: Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks. In: International Conference on Machine Learning (ICML) (2017)
21. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: International Conference on Learning Representations (ICLR) (2018)
22. Qi, G.J.: Loss-sensitive generative adversarial networks on Lipschitz densities. arXiv preprint arXiv:1701.06264 (2017)
23. Rabin, J., Peyré, G., Bernot, M.: Wasserstein barycenter and its application to texture mixing. In: International Conference on Scale Space and Variational Methods in Computer Vision (2011)
24. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: International Conference on Learning Representations (ICLR) (2016)
25. Rosca, M., Lakshminarayanan, B., Mohamed, S.: Distribution matching in variational inference. arXiv preprint arXiv:1802.06847 (2018)
26. Rosca, M., Lakshminarayanan, B., Warde-Farley, D., Mohamed, S.: Variational approaches for auto-encoding generative adversarial networks. arXiv preprint arXiv:1706.04987 (2017)
27. Roth, K., Lucchi, A., Nowozin, S., Hofmann, T.: Stabilizing training of generative adversarial networks through regularization. In: Advances in Neural Information Processing Systems (NIPS) (2017)
28. Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R.: Learning from simulated and unsupervised images through adversarial training. In: CVPR (2017)
29. Tabor, J., Knop, S., Spurek, P., Podolak, I., Mazur, M., Jastrzębski, S.: Cramer–Wold AutoEncoder. arXiv preprint arXiv:1805.09235 (2018)
30. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Adversarial generator-encoder networks. `https://github.com/DmitryUlyanov/AGE` (2018), gitHub repository
31. Ulyanov, D., Vedaldi, A., Lempitsky, V.: It takes (only) two: Adversarial generator-encoder networks. In: AAAI Conference on Artificial Intelligence (2018)
32. Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J.: LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365 (2015)
33. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.: StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: International Conference on Computer Vision (ICCV) (2017)
34. Zhou, W., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image qualifty assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (2008)

Supplementary Material for Pioneer Networks: Progressively Growing Generative Autoencoder

In this supplementary, we provide additional experiment figures that provide a broader overview on how the proposed method performs on CelebA and CelebA-HQ. We also include generated samples from the ALI and AGE methods.

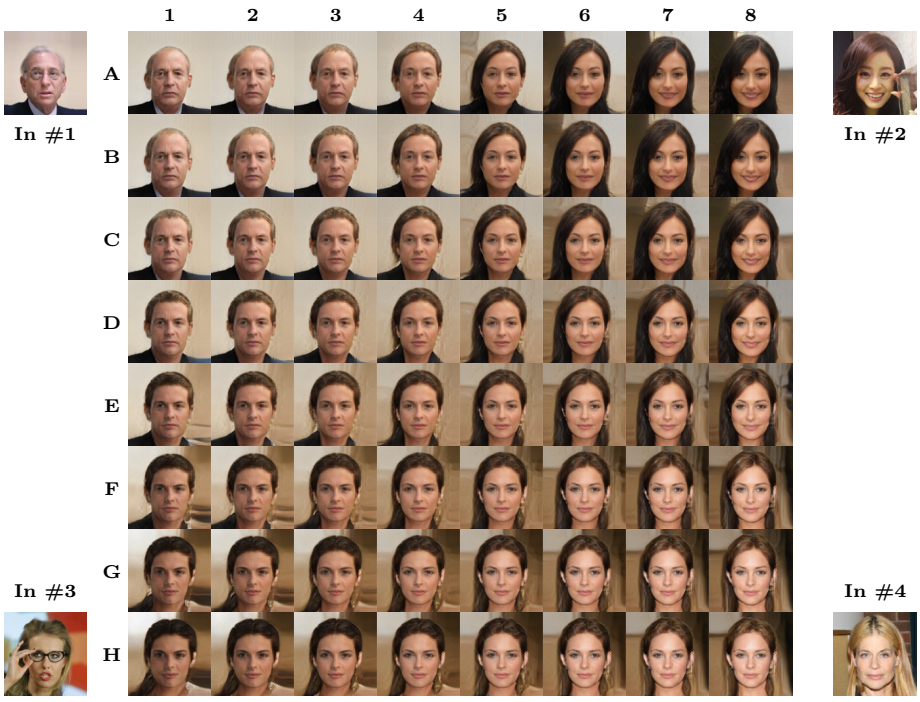


Fig. 8. Pioneer interpolation example on test set input images at 128×128 resolution.

Fig. 9. Examples of PIONEER network reconstruction (CELEBA) quality in 128×128 resolution.

Fig. 10. PIONEER random samples (CELEBA-HQ) at 256×256 resolution.

Fig. 11. PIONEER random samples (CELEBA) at 128×128 resolution.

Fig. 12. AGE random samples (CELEBA) at 128×128 resolution.

Fig. 13. AGE random samples (CelebA) at 64×64 resolution.

Fig. 14. ALI random samples (CelebA) at 64×64 resolution.

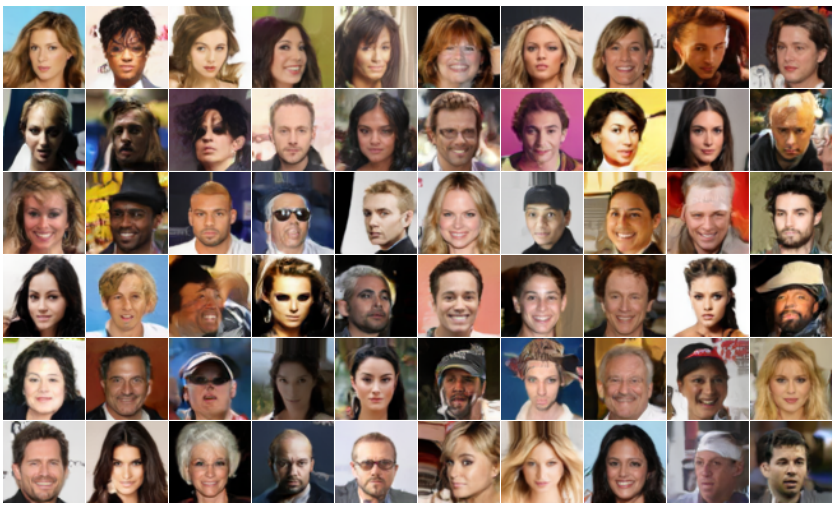


Fig. 15. PIONEER random samples (CELEBA) at 64×64 resolution.