Transferring GANs: generating images from limited data

Yaxing Wang, Chenshen Wu, Luis Herranz, Joost van de Weijer,
Abel Gonzalez-Garcia, Bogdan Raducanu
{yaxing, chenshen, lherranz, joost, agonzgarc, bogdan}@cvc.uab.es

Computer Vision Center
Universitat Autònoma de Barcelona, Spain

Abstract. Transferring knowledge of pre-trained networks to new domains by means of fine-tuning is a widely used practice for applications based on discriminative models. To the best of our knowledge this practice has not been studied within the context of generative deep networks. Therefore, we study domain adaptation applied to image generation with generative adversarial networks. We evaluate several aspects of domain adaptation, including the impact of target domain size, the relative distance between source and target domain, and the initialization of conditional GANs. Our results show that using knowledge from pre-trained networks can shorten the convergence time and can significantly improve the quality of the generated images, especially when target data is limited. We show that these conclusions can also be drawn for conditional GANs even when the pre-trained model was trained without conditioning. Our results also suggest that density is more important than diversity and a dataset with one or few densely sampled classes is a better source model than more diverse datasets such as ImageNet or Places.

Keywords: Generative adversarial networks, transfer learning, domain adaptation, image generation

1 Introduction

Generative Adversarial Networks (GANs) can generate samples from complex image distributions [1]. They consist of two networks: a discriminator which aims to separate real images from fake (or generated) images, and a generator which is simultaneously optimized to generate images which are classified as real by the discriminator. The theory was later extended to the case of conditional GANs where the generative process is constrained using a conditioning prior [2] which is provided as an additional input. GANs have further been widely applied in applications, including super-resolution [3], 3D object generation and reconstruction [4], human pose estimation [5], and age estimation [6].

Deep neural networks have obtained excellent results for discriminative classification problems for which large datasets exist; for example on the ImageNet dataset which consists of over 1M images [7]. However, for many problems the amount of labeled data is not sufficient to train the millions of parameters typically present in these networks. Fortunately, it was found that the knowledge contained in a network trained on a large dataset (such as ImageNet) can easily be transferred to other computer vision tasks. Either by using these networks as off-the-shelf feature extractors [8], or by adapting them to a new domain by a process called fine tuning [9]. In the latter case, the pre-trained network is used to initialize the weights for a new task (effectively transferring the knowledge learned from the source domain), which are then fine tuned with the training images from the new domain. It has been shown that much fewer images were required to train networks which were initialized with a pre-trained network.

GANs are in general trained from scratch. The procedure of using a pre-trained network for initialization – which is very popular for discriminative networks – is to the best of our knowledge not used for GANs. However, like in the case of discriminative networks, the number of parameters in a GAN is vast; for example the popular DC-GAN architecture [10] requires 36M parameters to generate an image of 64x64. Especially in the case of domains which lack many training images, the usage of pre-trained GANs could significantly improve the quality of the generated images.

Therefore, in this paper, we set out to evaluate the usage of pre-trained networks for GANs. The paper has the following contributions:

1. We evaluate several transfer configurations, and show that pre-trained networks can effectively accelerate the learning process and provide useful prior knowledge when data is limited.
2. We study how the relation between source and target domains impacts the results, and discuss the problem of choosing a suitable pre-trained model, which seems more difficult than in the case of discriminative tasks.
3. We evaluate the transfer from unconditional GANs to conditional GANs for two commonly used methods to condition GANs.

2 Related Work

Transfer learning/domain transfer: Learning how to transfer knowledge from a source domain to target domain is a well studied problem in computer vision [11]. In the deep learning era, complex knowledge is extracted during the training stage on large datasets [12, 13]. Domain adaptation by means of fine tuning a pre-trained network has become the default approach for many applications with limited training data or slow convergence [14, 9].

Several works have investigated transferring knowledge to unsupervised or sparsely labeled domains. Tzeng et al. [15] optimized for domain invariance, while transferring task information that is present in the correlation between the classes of the source domain. Ganin et al. [16] proposed to learn domain invariant features by means of a gradient reversal layer. A network simultaneously trained on these invariant features can be transfered to the target domain. Finally, domain transfer has also been studied for networks that learn metrics [17].

In contrast to these methods, we do not focus on transferring discriminative features, but transferring knowledge for image generation.

GAN: Goodfellow et al. [1] introduced the first GAN model for image generation. Their architecture uses a series of fully connected layers and thus is limited to simple datasets. When approaching the generation of real images of higher complexity, convolutional architectures have shown to be a more suitable option. Shortly afterwards, Deep Convolutional GANs (DC-GAN) quickly became the standard GAN architecture for image generation problems [10]. In DC-GAN, the generator sequentially up-samples the input features by using fractionally-strided convolutions, whereas the discriminator uses normal convolutions to classify the input images. Recent multi-scale architectures [18–20] can effectively generate high resolution images. It was also found that ensembles can be used to improve the quality of the generated distribution [21].

Independently of the type of architecture used, GANs present multiple challenges regarding their training, such as convergence properties, stability issues, or mode collapse. Arjovksy et al. [22] showed that the original GAN loss [1] are unable to properly deal with ill-suited distributions such as those with disjoint supports, often found during GAN training. Addressing these limitations the Wassertein GAN [23] uses the Wasserstein distance as a robust loss, yet requiring the generator to be 1-Lipschitz. This constrain is originally enforced by clipping the weights. Alternatively, an even more stable solution is adding a gradient penalty term to the loss (known as WGAN-GP) [24].

cGAN: Conditional GANs (cGANs) [2] are a class of GANs that use a particular attribute as a prior to build conditional generative models. Examples of conditions are class labels [25–27], text [28, 29], another image (image translation [30, 31] and style transfer [32]).

Most cGAN models [2, 29, 33, 34] apply their condition in both generator and discriminator by concatenating it to the input of the layers, i.e. the noise vector for the first layer or the learned features for the internal layers. Instead, in [32], they include the conditioning in the batch normalization layer. The AC-GAN framework [25] extends the discriminator with an auxiliary decoder to reconstruct class-conditional information. Similarly, InfoGAN [35] reconstructs a subset of the latent variables from which the samples were generated. Miyato et al. [36] propose another modification of the discriminator based on a projection layer that uses the inner product between the conditional information and the intermediate output to compute its loss.

3 Generative Adversarial Networks

3.1 Loss functions

A GAN consists of a generator G and a discriminator D [1]. The aim is to train a generator G which generates samples that are indistinguishable from the real data distribution. The discriminator is optimized to distinguish samples from the real data distribution p_{data} from those of the fake (generated) data distribution

p_g. The generator takes noise $z \sim p_z$ as input, and generates samples $G\left(z\right)$ with a distribution p_g. The networks are trained with an adversarial objective. The generator is optimized to generate samples which would be classified by the discriminator as belonging to the real data distribution. The minimax game objective is given by:

$$G^{*} = \underset{G}{\operatorname{argmin}} \max_{D} \mathcal{L}_{GAN}\left(G, D\right) \tag{1}$$

$$\mathcal{L}_{GAN}\left(G, D\right) = \mathbb{E}_{x \sim p_{data}}[\log D(x)] + \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z)))] \tag{2}$$

In the case of WGAN-GP [24] the two loss functions are:

$$\begin{aligned} \mathcal{L}_{WGAN-GP}\left(D\right) = & -\mathbb{E}_{x \sim p_{data}}[D(x)] + \mathbb{E}_{z \sim p_z}[D(G(z))] \\ & + \lambda \mathbb{E}_{x \sim p_{data}, z \sim p_z, \alpha \sim (0,1)} \left[\left(\|\nabla D\left(\alpha x + \left(1 - \alpha\right) G(z)\right)\|_2 - 1\right)^2 \right] \end{aligned} \tag{3}$$

$$\mathcal{L}_{WGAN-GP}\left(G\right) = -\mathbb{E}_{z \sim p_z}[D(G(z))] \tag{4}$$

3.2 Evaluation Metrics

Evaluating GANs is notoriously difficult [37] and there is no clear agreed reference metric yet. In general, a good metric should measure the quality and the diversity in the generated data. Likelihood has been shown to not correlate well with these requirements [37]. Better correlation with human perception has been found in the widely used Inception Score [38], but recent works have also shown its limitations [39]. In our experiments we use two recent metrics that show better correlation in recent studies [40, 41]. While not perfect, we believe they are satisfactory enough to help us to compare the models in our experiments.

Fréchet Inception Distance [42] The similarity between two sets is measured as their Fréchet distance (also known as Wasserstein-2 distance) in an embedded space. The embedding is computed using a fixed convolutional network (an Inception model) up to a specific layer. The embedded data is assumed to follow a multivariate normal distribution, which is estimated by computing their mean and covariance. In particular, the FID is computed as

$$\mathrm{FID}\left(\mathcal{X}_1, \mathcal{X}_2\right) = \|\mu_1 - \mu_2\|_2^2 + \mathrm{Tr}\left(\Sigma_1 + \Sigma_2 - 2\left(\Sigma_1 \Sigma_2\right)^{\frac{1}{2}}\right) \tag{5}$$

Typically, $\mathcal{X}_1$ is the full dataset with real images, while $\mathcal{X}_2$ is a set of generated samples. We use FID as our primary metric, since it is efficient to compute and correlates well with human perception [42].

Independent Wasserstein (IW) critic [43] This metric uses an independent critic $\hat{D}$ only for evaluation. This independent critic will approximate the Wasserstein distance [22] between two datasets $\mathcal{X}_1$ and $\mathcal{X}_2$ as

$$\mathrm{IW}\left(\mathcal{X}_1, \mathcal{X}_2\right) = \mathbb{E}_{x \sim \mathcal{X}_1}\left(\hat{D}\left(x\right)\right) - \mathbb{E}_{x \sim \mathcal{X}_2}\left(\hat{D}\left(x\right)\right) \tag{6}$$

Table 1: FID/IW (the lower the better / the higher the better) for different transfer configurations. ImageNet was used as source dataset and LSUN Bedrooms as target (100K images).

Generator	Scratch		Pre-trained	
Discriminator	Scratch	Pre-trained	Scratch	Pre-trained
FID $\left(\mathcal{X}^{tgt}_{data}, \mathcal{X}^{tgt}_{gen}\right)$	32.87	30.57	56.16	**24.35**
IW $\left(\mathcal{X}^{tgt}_{val}, \mathcal{X}^{tgt}_{gen}\right)$	-4.27	-4.02	-6.35	**-3.88**

In this case, $\mathcal{X}_1$ is typically a validation set, used to train the independent critic. We report IW only in some experiments, due to the larger computational cost that requires training a network for each measurement.

4 Transferring GAN representations

4.1 GAN adaptation

To study the effect of domain transfer for GANs we will use the WGAN-GP [24] architecture which uses ResNet in both generator and discriminator. This architecture has been experimentally demonstrated to be stable and robust against mode collapse [24]. The generator consists of one fully connected layer, four Residual Blocks and one convolution layer, and the Discriminator has same setting. The same architecture is used for conditional GAN.

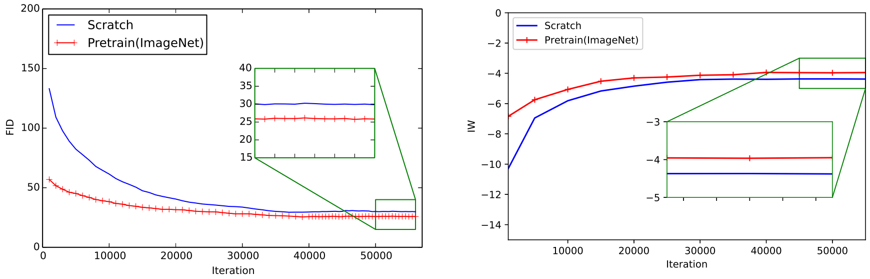
Implementation details We generate images of 64×64 pixels, using standard values for hyperparameters. The source models[1] are trained with a batch of 128 images during 50K iterations (except 10K iterations for CelebA) using Adam [44] and a learning rate of 1e-4. For fine tuning we use a batch size of 64 and a learning rate of 1e-4 (except 1e-5 for 1K target samples). Batch normalization and layer normalization are used in the generator and discriminator respectively.

4.2 Generator/discriminator transfer configuration

The two networks of the GAN (generator and discriminator) can be initialized with either random or pre-trained weights (from the source networks). In a first experiment we consider the four possible combinations using a GAN pre-trained with ImageNet and 100K samples of LSUN bedrooms as target dataset. The source GAN was trained for 50K iterations. The target GAN was trained for (additional) 40K iterations.

Table 1 shows the results. Interestingly, we found that transferring the discriminator is more critical than transferring the generator. The former helps to

[1] The pretrained models are available at https://github.com/yaxingwang/Transferring-GANs.



(a) Unconditional GAN(FID)

(b) Unconditional GAN(IW)

Fig. 1: Evolution of evaluation metrics when trained from scratch or using a pre-trained model for unconditional GAN measured with (a) FID and (b) IW (source: ImageNet, target: LSUN Bedrooms, metrics: FID and IW). The curves are smoothed for easier visualization by averaging in a window of a few iterations.

improve the results in both FID and IW metrics, while the latter only helps if the discriminator was already transferred, otherwise harming the performance. Transferring both obtains the best result. We also found that training is more stable in this setting. Therefore, in the rest of the experiments we evaluated either training both networks from scratch or pre-training both (henceforth simply referred to as *pre-trained*).

Figure 1 shows the evolution of FID and IW during the training process with and without transfer. Networks adapted from a pre-trained model can generate images of given scores in significantly fewer iterations. Training from scratch for a long time manages to reduce this gap significantly, but pre-trained GANs can generate images with good quality already with much fewer iterations. Figures 2 and 4 show specific examples illustrating visually these conclusions.

4.3 Size of the target dataset

The number of training images is critical to obtain realistic images, in particular as the resolution increases. Our experimental settings involve generating images of 64×64 pixels, where GANs typically require hundreds of thousands of training images to obtain convincing results. We evaluate our approach in a challenging setting where we use as few as 1000 images from the LSUN Bedrooms dataset, and using ImageNet as source dataset. Note that, in general, GANs evaluated on LSUN Bedrooms use the full set of 3M million images.

Table 2 shows FID and IW measured for different amounts of training samples of the target domain. As the training data becomes scarce, the training set implicitly becomes less representative of the full dataset (i.e. less diverse). In this experiment, a GAN adapted from the pre-trained model requires roughly between two and five times fewer images to obtain a similar score than a GAN trained from scratch. FID and IW are sensitive to this factor, so in order to

Table 2: FID/IW for different sizes of the target set (LSUN Bedrooms) using ImageNet as source dataset.

Target samples	1K	5K	10K	50K	100K	500K	1M
From scratch	256.1/-33.3	86.0/-18.5	73.7/-15.3	45.5/-7.4	32.9/-4.3	24.9/-3.6	21.0/-2.9
Pre-trained	93.4/-22.5	74.3/-16.3	47.0/-7.0	29.6/-4.56	24.4/-4.0	21.6/-3.2	18.5/-2.8

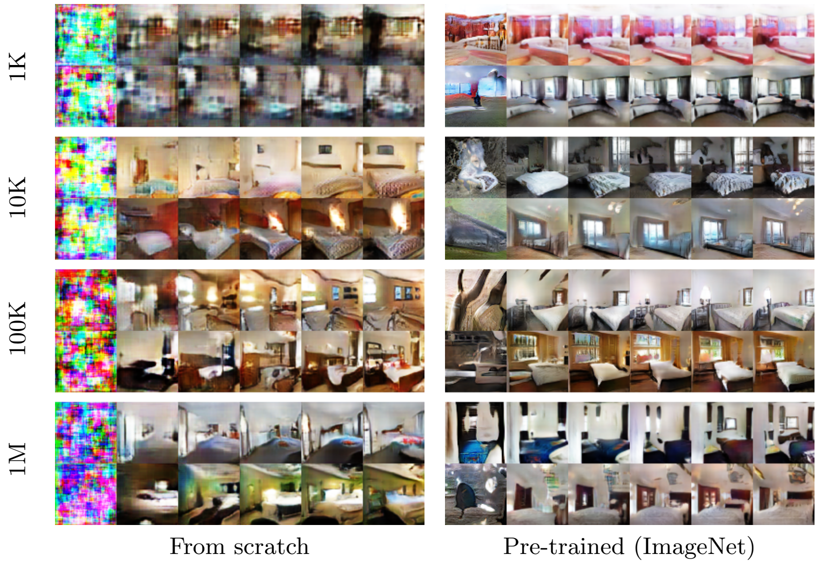


Fig. 2: Images generated at different iterations (from 0 and 10000, step 2000) for LSUN bedrooms training from scratch and from a pre-trained network. Better viewed in electronic version.

have a lower bound we also measured the FID between the specific subset used as training data and the full dataset. With 1K images this value is even higher than the value for generated samples after training with 100K and 1M images.

Intializing with the pre-trained GAN helps to improve the results in all cases, being more significant as the target data is more limited. The difference with the lower bound is still large, which suggests that there is still field for improvement in settings with limited data.

Figure 2 shows images generated at different iterations. As in the previous case, pre-trained networks can generate high quality images already in earlier iterations, in particular with sharper and more defined shapes and more realistic fine details. Visually, the difference is also more evident with limited data, where learning to generate fine details is difficult, so adapting pre-trained networks can transfer relevant prior information.

Table 3: Datasets used in the experiments.

Source datasets	ImageNet [12]	Places [13]	Bedrooms [45]	CelebA [46]
Number of images	1M	2.4M	3M	200K
Number of classes	1000	205	1	1
Target datasets	Flower [47]	Kitchens [45]	LFW [48]	Cityscapes [49]
Number of images	8K	50K	13K	3.5K
Number of classes	102	1	1	1

Table 4: Distance between target real data and target generated data FID/IW $\left(\mathcal{X}^{tgt}_{data}, \mathcal{X}^{tgt}_{gen}\right)$.

Source → Target ↓	Scratch	ImageNet	Places	Bedrooms	CelebA
Flowers	71.98/-13.62	**54.04/-3.09**	66.25/-5.97	56.12/-5.90	67.96/-12.64
Kitchens	42.43/-7.79	34.35/-4.45	34.59/**-2.92**	**28.54**/-3.06	38.41/-4.98
LFW	19.36/-8.62	9.65/-5.17	15.02/-6.61	7.45/-3.61	**7.16/-3.45**
Cityscapes	155.68/-9.32	**122.46**/-9.00	151.34/-8.94	123.21/-8.44	130.64/**-6.40**

4.4 Source and target domains

The domain of the source model and its relation with the target domain are also a critical factor. We evaluate different combinations of source domains and target domains (see Table 3 for details). As source datasets we used ImageNet, Places, LSUN Bedrooms and CelebA. Note that both ImageNet and Places cover wide domains, with great diversity in objects and scenes, respectively, while LSUN Bedrooms and CelebA cover more densely a narrow domain. As target we used smaller datasets, including Oxford Flowers, LSUN Kitchens (a subset of 50K out of 2M images), Label Faces in the Wild (LFW) and CityScapes.

We pre-trained GANs for the four source datasets and then trained five GANs for each of the four target datasets (from scratch and initialized with each of the source GANs). The FID and IW after fine tuning are shown in Table 4. Pre-trained GANs achieve significantly better results. Both metrics generally agree but there are some interesting exceptions. The best source model for Flowers as target is ImageNet, which is not surprising since it contains also flowers, plants and objects in general. It is more surprising that Bedrooms is also competitive according to FID (but not so much according to IW). The most interesting case is perhaps Kitchens, since Places has several thousands of kitchens in the dataset, yet also many more classes that are less related. In contrast, bedrooms and kitchens are not the same class yet still very related visually and structurally, so the much larger set of related images in Bedrooms may be a better choice. Here FID and IW do not agree, with FID clearly favoring Bedrooms, and even the less related ImageNet, over Places, while IW preferring Places by a small

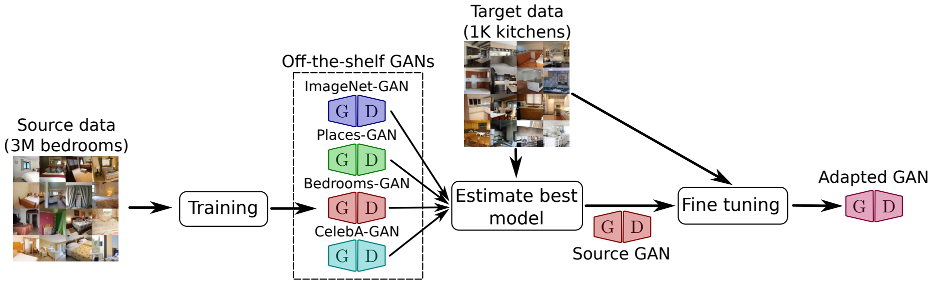


Fig. 3: Transferring GANs: training source GANs, estimation of the most suitable pre-trained model and adaptation to the target domain.

Table 5: Distance between source generated data $\mathcal{X}^{src}_{gen}$ and target real data $\mathcal{X}^{tgt}_{data}$, and distance between source real $\mathcal{X}^{src}_{data}$ and generated data $\mathcal{X}^{src}_{gen}$.

	Source → Target ↓	ImageNet	Places	Bedrooms	CelebA
FID $\left(\mathcal{X}^{src}_{gen}, \mathcal{X}^{tgt}_{data}\right)$	Flowers	**237.04**	251.93	278.80	284.74
	Kitchens	183.27	180.63	**70.06**	254.12
	LFW	333.54	333.38	329.92	**151.46**
	Cityscapes	233.45	**181.72**	227.53	292.66
FID $\left(\mathcal{X}^{src}_{gen}, \mathcal{X}^{src}_{data}\right)$	Source	63.46	55.66	17.30	75.84

margin. As expected, CelebA is the best source for LFW, since both contain faces (with different scales though), but Bedroom is surprisingly very close to the performance in both metrics. For Cityscapes all methods have similar results (within a similar range), with both high FID and IW, perhaps due to the large distance to all source domains.

4.5 Selecting the pre-trained model

Selecting a pre-trained model for a discriminative task (e.g. classification) is reduced to simply selecting either ImageNet, for object-centric domains, or Places, for scene-centric ones. The target classifier or fine tuning will simply learn to ignore non-related features and filters of the source network.

However, this simple rule of thumb does not seem to apply so clearly in our GAN transfer setting due to generation being a much more complex task than discrimination. Results in Table 4 show that sometimes unrelated datasets may perform better than other apparently more related. The large number of unrelated classes may be an important factor, since narrow yet dense domains also seem to perform better even when they are not so related (e.g. Bedrooms). There are also non-trivial biases in the datasets that may explain this behavior. Therefore, a way to estimate the most suitable model for a given target dataset is desirable, given a collection of pre-trained GANs.

Perhaps the most simple way is to measure the distance between the source and target domains. We evaluated the FID between the (real) images in the target and the source datasets (results included in the supplementary material). While showing some correlation with the FID of the target generated data, it has the limitation of not considering whether the actual pre-trained model is able or not to accurately sample from the real distribution. A more helpful metric is the distance between the target data and the *generated* samples by the pre-trained model. In this way, the quality of the model is taken into account. We estimate this distance also using FID. In general, there seem to roughly correlate with the final FID results with target generated data (compare Tables 4 and 5). Nevertheless, it is surprising that Places is estimated as a good source dataset but does not live up to the expectation. The opposite occurs for Bedrooms, which seems to deliver better results than expected. This may suggest that density is more important than diversity for a good transferable model, even for apparently unrelated target domains.

In our opinion, the FID between source generated and target real data is a rough indicator of suitability rather than accurate metric. It should taken into account jointly with others factors (e.g. quality of the source model) to decide which model is best for a given target dataset.

4.6 Visualizing the adaptation process

One advantage of the image generation setting is that the process of shifting from the source domain towards the target domain can be visualized by sampling images at different iterations, in particular during the initial ones. Figure 4 shows some examples of the target domain Kitchens and different source domains (iterations are sampled in a logarithmic scale).

Trained from scratch, the generated images simply start with noisy patterns that evolve slowly, and after 4000 iterations the model manages to reproduce the global layout and color, but still fails to generate convincing details. Both the GANs pre-trained with Places and ImageNet fail to generate realistic enough source images and often sample from unrelated source classes (see iteration 0). During the initial adaptation steps, the GAN tries to generate kitchen-like patterns by matching and slightly modifying the source pattern, therefore preserving global features such as colors and global layout, at least during a significant number of iterations, then slowly changing them to more realistic ones. Nevertheless, the textures and edges are sharper and more realistic than from scratch. The GAN pre-trained with Bedrooms can already generate very convincing bedrooms, which share a lot of features with kitchens. The larger number of training images in Bedrooms helps to learn transferable fine grained details that other datasets cannot. The adaptation mostly preserves the layout, colors and perspective of the source generated bedroom, and slowly transforms it into kitchens by changing fine grained details, resulting in more convincing images than with the other source datasets. Despite being a completely unrelated domain, CelebA also manages to help in speeding up the learning process by providing useful priors. Different parts such as face, hair and eyes are transformed into different parts of the kitchen. Rather than the face itself, the most predominant feature remaining from the source generated image is the background color and shape, that influences in the layout and colors that the generated kitchens will have.

5 Transferring to conditional GANs

Here we study the transferring the representation learned by a pre-trained unconditional GAN to a cGAN [2]. cGANs allow us to condition the generative model on particular information such as classes, attributes, or even other images. Let y be a conditioning variable. The discriminator $D(x, y)$ aims to distinguish pairs of real data x and y sampled from the joint distribution $p_{data}\left(x, y\right)$ from pairs of generated outputs $G(z, y')$ conditioned on samples y' from y's marginal $p_{data}(y)$.

5.1 Conditional GAN adaptation

For the current study, we adopt the Auxiliary Classifier GAN (AC-GAN) framework of [25]. In this formulation, the discriminator has an ‘auxiliary classifier’ that outputs a probability distribution over classes $P(C = y|x)$ conditioned on the input x. The objective function is then composed of the conditional version of the GAN loss $\mathcal{L}_{GAN}$ (eq. (2)) and the log-likelihood of the correct class. The final loss functions for generator and discriminator are:

$$\mathcal{L}_{AC-GAN}\left(G\right) = \mathcal{L}_{GAN}\left(G\right) - \alpha_G \mathbb{E}\left[\log\left(P\left(C = y'|G(z, y')\right)\right)\right], \tag{7}$$

$$\mathcal{L}_{AC-GAN}\left(D\right) = \mathcal{L}_{GAN}\left(D\right) - \alpha_D \mathbb{E}\left[\log\left(P\left(C = y|x\right)\right)\right], \tag{8}$$

respectively. The parameters α_G and α_D weight the contribution of the auxiliary classifier loss with respect to the GAN loss for the generator and discriminator. In our implementation, we use Resnet-18 [50] for both G and D, and the WGAN-GP loss from the equations (3) and (4) as the GAN loss. Overall, the implementation details (batch size, learning rate) are the same as introduced in section 4.1.

In AC-GAN, the conditioning is performed only on the generator by appending the class label to the input noise vector. We call this variant ‘Cond Concat’. We randomly initialize the weights which are connected to the conditioning prior. We also used another variant following [32], in which the conditioning prior is embedded in the batch normalization layers of the generator (referred to as ‘Cond BNorm’). In this case, there are different batch normalization parameters for each class. We initialize these parameters by copying the values from the unconditional GAN to all classes.

5.2 Results

We use Places [13] as the source domain and consider all the ten classes of the LSUN dataset [45] as target domain. We train the AC-GAN with 10K images per class for 25K iterations. The weights of the conditional GAN can be transferred from the pre-trained unconditional GAN (see section 3.1) or initialized at random. The performance is assessed in terms of the FID score between target domain and generated images. The FID is computed class-wise, averaging

Table 6: Per-class and overall FID for AC-GAN. Source: Places, target: LSUN

Init	Iter	Bedr	Bridge	Church	Classr	Confer	Dining	Kitchen	Living	Rest	Tower	Avg.	All
Scratch	250	298.4	310.3	314.4	376.6	339.1	294.9	314.2	316.5	324.4	301.0	319.0	352.4
	2500	195.9	135.0	133.0	218.6	185.3	173.9	167.9	189.3	159.5	125.6	168.4	137.3
	25000	72.9	78.0	52.4	106.7	76.9	40.1	53.9	56.1	74.7	59.8	67.2	49.6
Pre-trained	250	**168.3**	**122.1**	**148.1**	**145.0**	**151.6**	**144.2**	**156.9**	**150.1**	**113.3**	**129.7**	**142.9**	**107.2**
	2500	**140.8**	**96.8**	**77.4**	**136.0**	**136.8**	**84.6**	**85.5**	**94.9**	**77.0**	**69.4**	**99.9**	**74.8**
	25000	**59.9**	**68.6**	**48.2**	**79.0**	**68.7**	**35.2**	**48.2**	**47.9**	**44.4**	**49.9**	**55.0**	**42.7**

over all classes and also considering the dataset as a whole (class-agnostic case). The classes in the target domain have been generated uniformly. The results are presented in table 6, where we show the performance of the AC-GAN whose weights have been transferred from pre-trained network vs. an AC-GAN initialized randomly. We computed the FID for 250, 2500 and 25000 iterations. At the beginning of the learning process, there is a significant difference between the two cases. The gap is reduced towards the end of the learning process but a significant performance gain still remains for pre-trained networks. We also consider the case with fewer images per class. The results after 25000 iterations for 100 and 1K images per class are provided in the last column of table 7. We can observe how the difference between networks trained from scratch or from pre-trained weights is more significant for smaller sample sizes. This confirms the trend observed in section 4.3: transferring the pre-trained weights is especially advantageous when only limited data is available.

The same behavior can be observed in figure 5 (left) where we compare the performance of the AC-GAN with two unconditional GANs, one pre-trained on the source domain and one trained from scratch, as in section 4.2. The curves correspond to the class-agnostic case (column 'All' in the table 6). From this plot, we can observe three aspects: (i) the two variants of AC-GAN perform similarly (for this reason, for the remaining of the experiments we consider only 'Cond BNorm'); (ii) the network initialized with pre-trained weights converges faster than the network trained from scratch, and the overall performance is better; and (iii) AC-GAN performs slightly better than the unconditional GAN.

Next, we evaluate the AC-GAN performance on a classification experiment. We train a reference classifier on the 10 classes of LSUN (10K real images per class). Then, we evaluate the quality of each model trained for 25K iterations by generating 10K images per class and measuring the accuracy of the reference classifier for 100, 1K and 10K images per class. The results show an improvement when using pre-trained models, with higher accuracy and lower FID in all settings, suggesting that it captures better the real data distribution of the dataset compared to training from scratch.

Finally, we perform a psychophysical experiment with generated images by AC-GAN with LSUN as target. Human subjects are presented with two images: pre-trained vs. from scratch (generated from the same condition `<class>`), and asked 'Which of these two images of `<class>` is more realistic?' Subjects were also given the option to skip a particular pair should they find very hard to decide for one of them. We require each subject to provide 100 valid assessments. We use 10 human subjects which evaluate image pairs for different settings (100, 1K,

Table 7: Accuracy of AC-GAN for the classification task and overall FID for different sizes of the target set (LSUN).

#images	Method	Accuracy (%)											FID
		Bedr	Bridge	Church	Classr	Confer	Dining	Kitchen	Living	Rest	Tower	Avg.	
100/class	scratch	23.0	**88.2**	**55.1**	29.2	3.6	24.9	20.8	8.4	**89.3**	**61.6**	40.4	162.9
	pre-trained	**35.7**	72.7	45.7	**59.4**	**7.9**	**38.2**	**36.3**	**20.1**	81.0	56.6	**45.4**	**119.1**
1K/class	scratch	49.9	78.1	**75.1**	51.8	14.6	51.2	31.2	23.2	**90.7**	61.5	52.7	117.3
	pre-trained	**76.4**	**82.5**	69.1	**80.6**	**34.2**	**52.6**	**62.4**	**52.9**	80.5	**67.5**	**65.9**	**77.5**
10K/class	scratch	**94.9**	94.3	89.6	85.0	82.4	**91.2**	88.0	86.9	91.3	83.5	88.7	49.6
	pre-trained	87.1	**95.7**	**90.8**	**95.1**	**86.8**	90.2	**88.9**	**90.1**	**93.0**	**88.9**	**90.8**	**42.7**

10K images per class). The results (Fig. 5 right) clearly show that the images based on pre-trained GANs are considered to be more realistic in the case of 100 and 1K images per class (e.g. pre-trained is preferred in 67% of cases with 1K images). As expected the difference is smaller for the 10K case.

6 Conclusions

We show how the principles of transfer learning can be applied to generative features for image generation with GANs. GANs, and conditional GANs, benefit from transferring pre-trained models, resulting in lower FID scores and more recognizable images with less training data. Somewhat contrary to intuition, our experiments show that transferring the discriminator is much more critical than the generator (yet transferring both networks is best). However, there are also other important differences with the discriminative scenario. Notably, it seems that a much higher density (images per class) is required to learn good transferable features for image generation, than for image discrimination (where diversity seems more critical). As a consequence, ImageNet and Places, while producing excellent transferable features for discrimination, seem not dense enough for generation, and LSUN data seems to be a better choice despite its limited diversity. Nevertheless, poor transferability may be also related to the limitations of current GAN techniques, and better ones could also lead to better transferability.

Our experiments evaluate GANs in settings rarely explored in previous works and show that there are many open problems. These settings include GANs and evaluation metrics in the very limited data regime, better mechanisms to estimate the most suitable pre-trained model for a given target dataset, and the design of better pre-trained GAN models.

Acknowledgement Y. Wang and C. Wu acknowledge the Chinese Scholarship Council (CSC) grant No.201507040048 and No.201709110103. L. Herranz acknowledges the European Union research and innovation program under the Marie Skodowska-Curie grant agreement No. 6655919. This work was supported by TIN2016-79717-R, and the CHISTERA project M2CR (PCIN-2015-251) of the Spanish Ministry, the CERCA Program of the *Generalitat de Catalunya*. We also acknowledge the generous GPU support from NVIDIA.

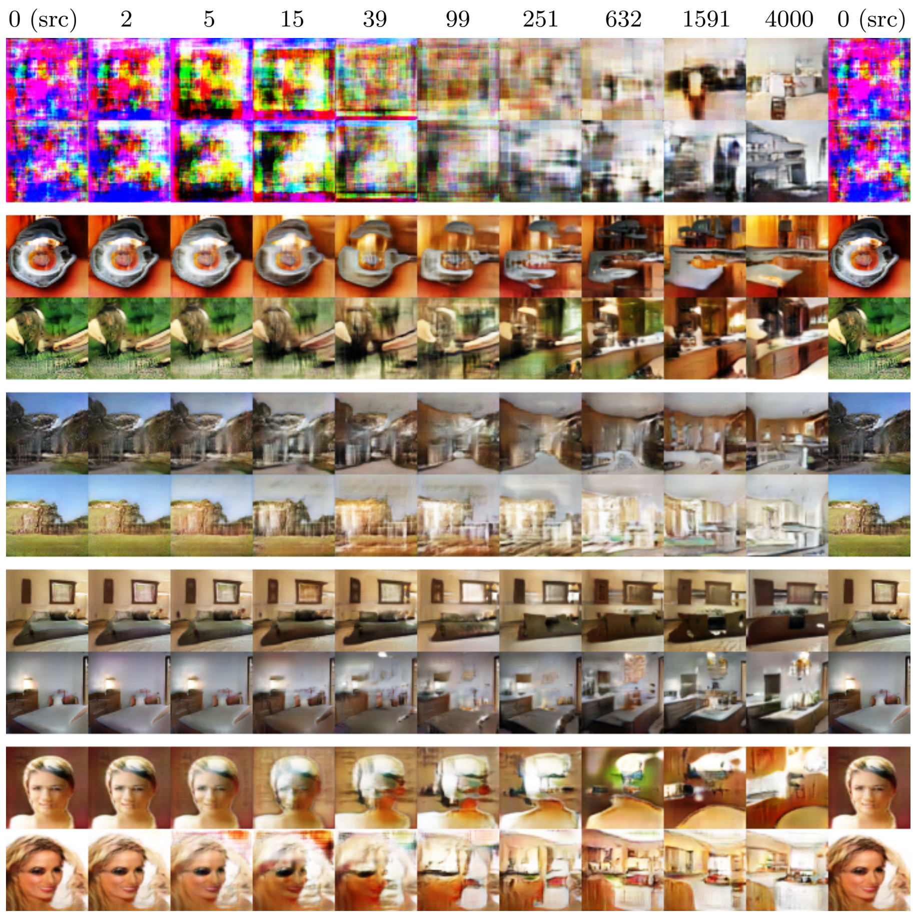


Fig. 4: Evolution of generated images (in logarithmic scale) for LSUN kitchens with different source datasets (from top to bottom: from scratch, ImageNet, Places, LSUN bedrooms, CelebA). Better viewed in electronic version.

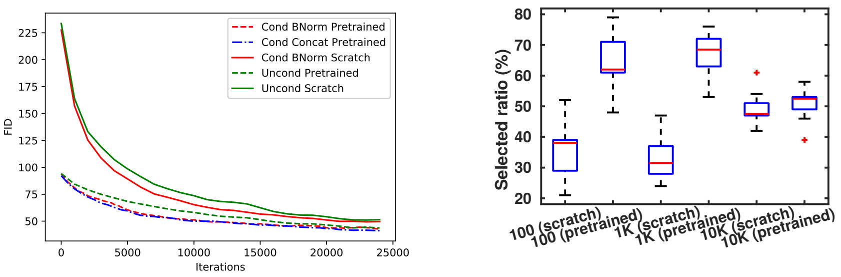


Fig. 5: (Left) FID score for Conditional and Unconditional GAN (source: Places, target: LSUN 10 classes). (Right) Human evaluation of image quality.

References

1. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS. (2014) 2672–2680
2. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
3. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR. (2016)
4. Smith, E., Meger, D.: Improved adversarial systems for 3d object generation and reconstruction. arXiv preprint arXiv:1707.09557 (2017)
5. Ma, L., Jia, X., Sun, Q., Schiele, B., Tuytelaars, T., Van Gool, L.: Pose guided person image generation. In: NIPS. (2017) 405–415
6. Zhang, Z., Song, Y., Qi, H.: Age progression/regression by conditional adversarial autoencoder. In: CVPR. Volume 2. (2017)
7. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. (2012) 1097–1105
8. Azizpour, H., Razavian, A.S., Sullivan, J., Maki, A., Carlsson, S.: Factors of transferability for a generic convnet representation. IEEE Trans. on PAMI **38**(9) (2016) 1790–1802
9. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: CVPR, IEEE (2014) 1717–1724
10. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: ICLR. (2015)
11. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering **22**(10) (2010) 1345–1359
12. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. IJCV **115**(3) (2015) 211–252
13. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: NIPS. (2014) 487–495
14. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. In: ICML. (2014) 647–655
15. Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Simultaneous deep transfer across domains and tasks. In: CVPR. (2015) 4068–4076
16. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. JMLR **17**(1) (2016) 2096–2030
17. Hu, J., Lu, J., Tan, Y.P.: Deep transfer metric learning. In: CVPR, IEEE (2015) 325–333
18. Denton, E.L., Chintala, S., Fergus, R., et al.: Deep generative image models using a laplacian pyramid of adversarial networks. In: NIPS. (2015) 1486–1494
19. Huang, X., Li, Y., Poursaeed, O., Hopcroft, J., Belongie, S.: Stacked generative adversarial networks. In: CVPR. Volume 2. (2017) 4
20. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. In: ICLR. (2018)
21. Wang, Y., Zhang, L., van de Weijer, J.: Ensembles of generative adversarial networks. In: NIPS 2016 Workshop on Adversarial Training. (2016)

22. Arjovsky, M., Bottou, L.: Towards principled methods for training generative adversarial networks. In: ICLR. (2017)
23. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: ICML. (2017) 214–223
24. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: NIPS. (2017) 5769–5779
25. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier gans. In: ICML. (2016)
26. Perarnau, G., van de Weijer, J., Raducanu, B., Álvarez, J.M.: Invertible conditional gans for image editing. In: NIPS 2016 Workshop on Adversarial Training. (2016)
27. Grinblat, G.L., Uzal, L.C., Granitto, P.M.: Class-splitting generative adversarial networks. arXiv preprint arXiv:1709.07359 (2017)
28. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. In: ICML. (2016) 1060–1069
29. Zhang, H., Xu, T., Li, H., Zhang, S., Huang, X., Wang, X., Metaxas, D.: Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: ICCV. (2017) 5908–5916
30. Kim, T., Cha, M., Kim, H., Lee, J., Kim, J.: Learning to discover cross-domain relations with generative adversarial networks. In: ICML. (2017) 1857–1865
31. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV. (2017) 2242–2251
32. Dumoulin, V., Shlens, J., Kudlur, M.: A learned representation for artistic style. In: ICLR. (2017)
33. Dumoulin, V., Belghazi, I., Poole, B., Lamb, A., Arjovsky, M., Mastropietro, O., Courville, A.: Adversarially learned inference. In: ICLR. (2017)
34. Sricharan, K., Bala, R., Shreve, M., Ding, H., Saketh, K., Sun, J.: Semi-supervised conditional gans. arXiv preprint arXiv:1708.05789 (2017)
35. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In: NIPS. (2016)
36. Miyato, T., Koyama, M.: Cgans with projection discriminator. In: ICLR. (2018)
37. Theis, L., Oord, A.v.d., Bethge, M.: A note on the evaluation of generative models. In: ICLR. (2015)
38. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: NIPS. (2016) 2234–2242
39. Zhou, Z., Cai, H., Rong, S., Song, Y., Ren, K., Zhang, W., Yu, Y., Wang, J.: Activation maximization generative adversarial nets. In: ICLR. (2018)
40. Im, D.J., Ma, H., Taylor, G., Branson, K.: Quantitatively evaluating gans with divergences proposed for training. In: ICLR. (2018)
41. Borji, A.: Pros and cons of gan evaluation measures. arXiv preprint arXiv:1802.03446 (2018)
42. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Klambauer, G., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a nash equilibrium. In: NIPS. (2017)
43. Danihelka, I., Lakshminarayanan, B., Uria, B., Wierstra, D., Dayan, P.: Comparison of maximum likelihood and gan-based training of real nvps. arXiv preprint arXiv:1705.05263 (2017)
44. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR. (2014)

45. Yu, F., Zhang, Y., Song, S., Seff, A., Xiao, J.: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365 (2015)
46. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: ICCV. (2015) 3730–3738
47. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: ICVGIP, IEEE (2008) 722–729
48. Huang, G.B., Mattar, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In: Workshop on faces in'Real-Life'Images: detection, alignment, and recognition. (2008)
49. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR. (2016) 3213–3223
50. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016) 770–778

Supplementary Material

A Distances between source and target data

Table A1 shows the FID between the real images in the source and target datasets, which could be used as an estimation of which pre-trained GAN (on a source dataset) may be a good choice to adapt to a particular target dataset. In most of the cases, the lowest value in Table A1 also corresponds to the lowest value in Table 1.

Table A1: Distance between source real data and target real data.

Distance	Source → Target ↓	ImageNet	Places	Bedrooms	CelebA
FID $\left(\mathcal{X}_{data}^{src}, \mathcal{X}_{data}^{tgt}\right)$	Flowers	**187.52**	292.36	270.09	317.21
	Kitchens	139.81	99.88	**66.54**	311.06
	LFW	266.50	326.76	318.98	**44.12**
	Cityscapes	205.04	**143.55**	221.65	349.28

B Model capacity

In order to check how important the capacity of the network is for transferring GAN features, we performed an additional experiment where we reduced the capacity of the network to half. We trained a source GAN with ImageNet, but in this case we reduced the number of filters in each layer to half its original value (with respect to the architecture used throughout our paper, from WGAN-GP [25]). The model is then fine tuned with 10K images from LSUN Bedrooms. The results shown in Fig. 6 suggest that also a lower capacity GAN adapting pre-trained features obtains significantly better results.

C Images sampled from the models

We also show examples of images sampled from each of the source models after fine tuning 5K iterations with Flowers (Fig. 7), Kitchens (Fig. 8), LFW (Fig. 9), and cityscapes (Fig. 10).

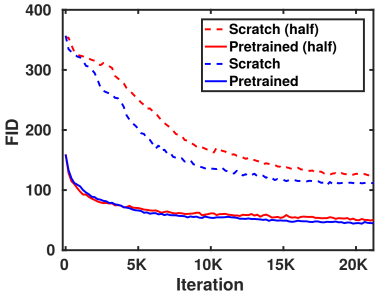


Fig. 6: Model capacity.

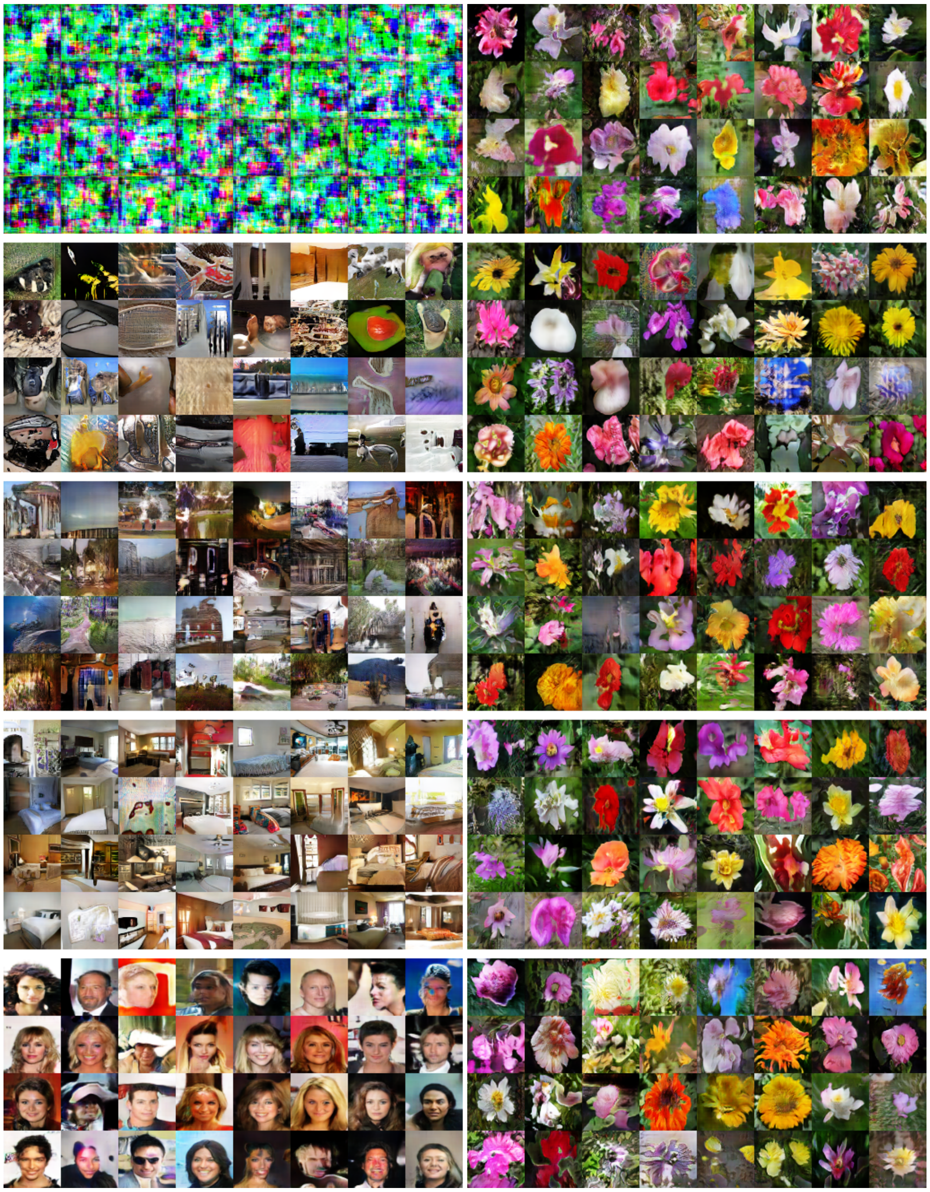


Fig. 7: Images sampled from each of the source models (left) and after fine tuning 5K iterations with Flowers (right). From top to bottom: from scratch, ImageNet, Places, LSUN bedrooms, CelebA.

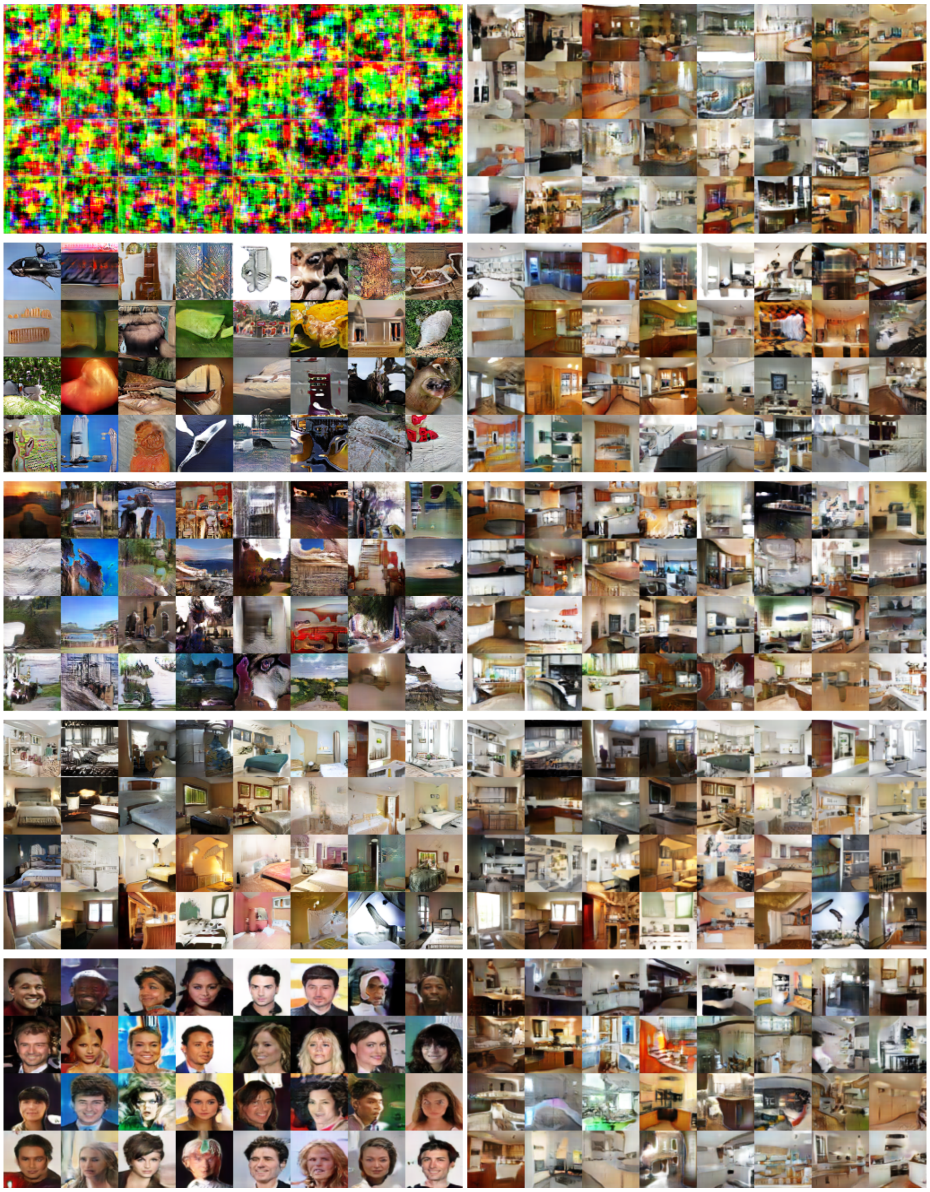


Fig. 8: Images sampled from each of the source models (left) and after fine tuning 5K iterations with Kitchens (right). From top to bottom: from scratch, ImageNet, Places, LSUN bedrooms, CelebA.

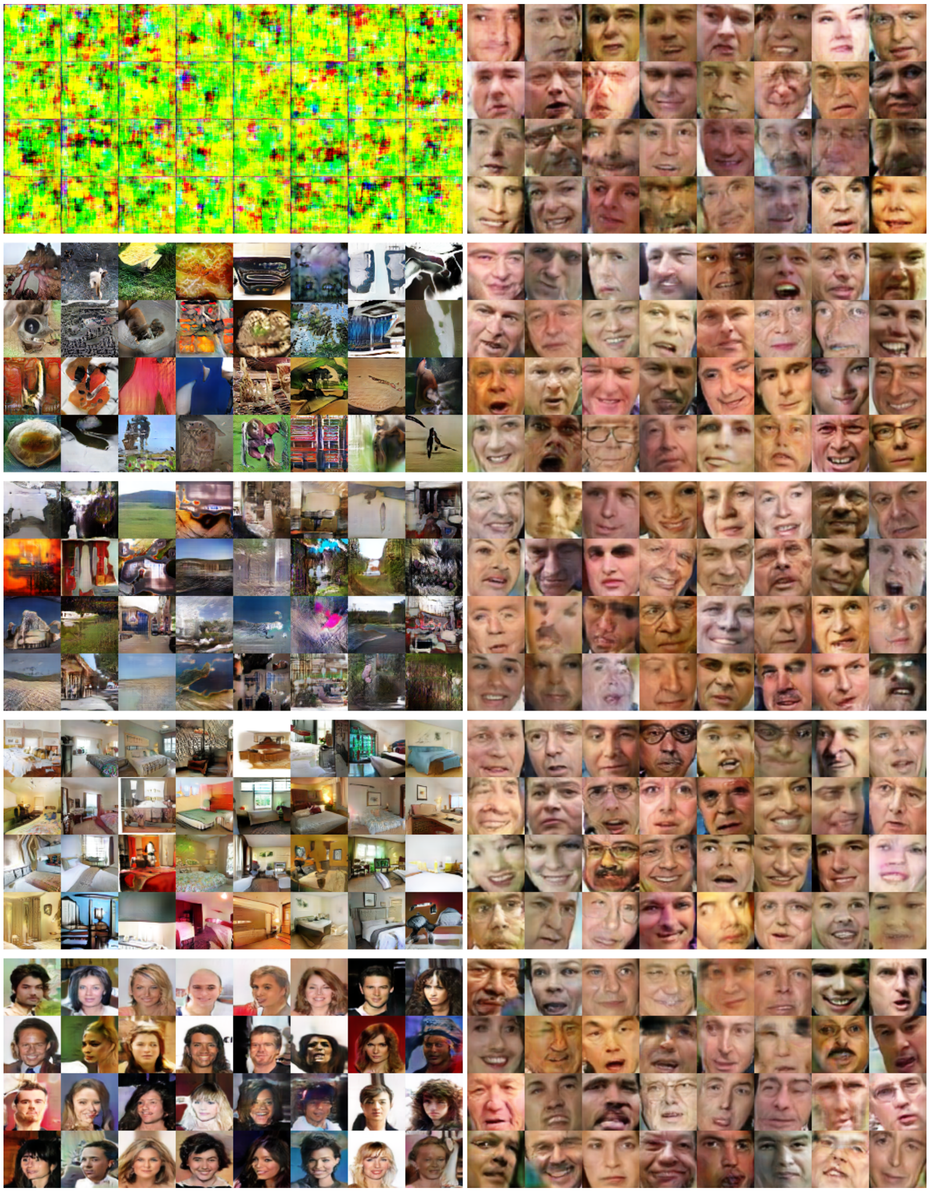


Fig. 9: Images sampled from each of the source models (left) and after fine tuning 5K iterations with LFW (right). From top to bottom: from scratch, ImageNet, Places, LSUN bedrooms, CelebA.

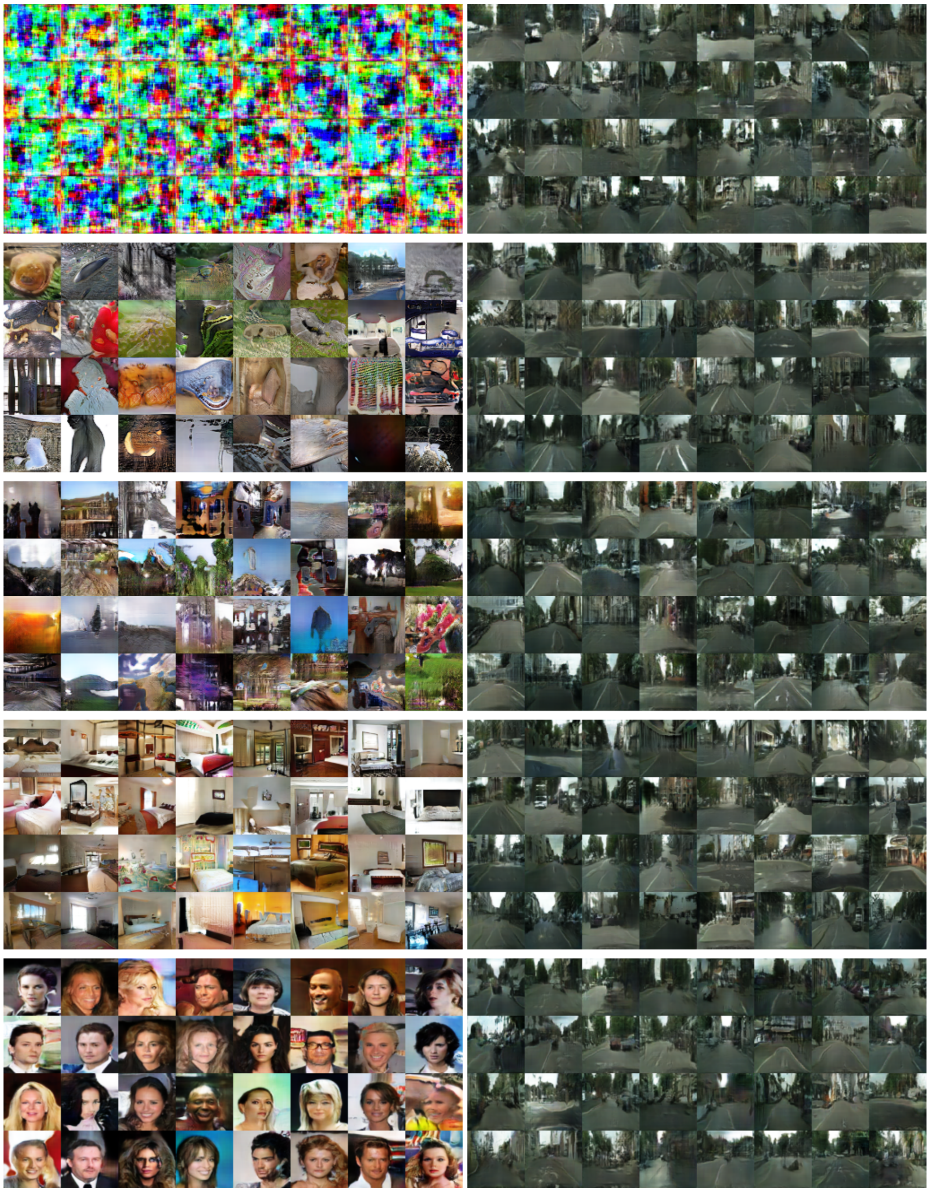


Fig. 10: Images sampled from each of the source models (left) and after fine tuning 5K iterations with Cityscapes (right). From top to bottom: from scratch, ImageNet, Places, LSUN bedrooms, CelebA.