FewGAN: GENERATING FROM THE JOINT DISTRIBUTION OF A FEW IMAGES

Lior Ben-Moshe[1] *Sagie Benaim*[2] *Lior Wolf*[1]

[1]Tel Aviv University, [2]University of Copenhagen

ABSTRACT

We introduce FewGAN, a generative model for generating novel, high-quality and diverse images whose patch distribution lies in the joint patch distribution of a small number of $N > 1$ training samples. The method is, in essence, a hierarchical patch-GAN that applies quantization at the first coarse scale, in a similar fashion to VQ-GAN, followed by a pyramid of residual fully convolutional GANs at finer scales. Our key idea is to first use quantization to learn a fixed set of patch embeddings for training images. We then use a separate set of side images to model the structure of generated images using an autoregressive model trained on the learned patch embeddings of training images. Using quantization at the coarsest scale allows the model to generate both conditional and unconditional novel images. Subsequently, a patch-GAN renders the fine details, resulting in high-quality images. In an extensive set of experiments, it is shown that FewGAN outperforms baselines both quantitatively and qualitatively.

Index Terms— GANs, Few-Shot learning, Quantization

1. INTRODUCTION

We present a generative model for generating novel, high-quality, and diverse samples whose patch distribution lies in the joint patch distribution of a small number of training samples, in a coherent manner. Our method is capable of generating both conditional and unconditional images, as well as solving a variety of image manipulation tasks, including editing, inpainting and harmonization. In all of these cases, our model produces high-quality results that preserve the internal patch statistics of the training images. All tasks are achieved with the same generative network, without any further training.

Generating a coherent and realistic image, in an unsupervised manner, from the joint patch distribution of a small number of $N > 1$ samples is a challenging problem. First, the joint patch distribution is not evident in any of the training images, yet we are interested in generating a realistic image that depicts patches from multiple images simultaneously in the same image. Second, one has to avoid both a mode collapse in which all patches are taken from the same input image and the mode collapse of generating an image that is very similar to a single training image.

As a main tool for addressing these two issues, we model the patch distribution of training images using a vector-quantized (VQ) basis learned on the N training images. We then use a side dataset of unlabeled images, taken from random sources (naturally, there is no shortage of such images) to train an autoregressive model that is able to combine patches that have originated from the training images, but preserve the structure of images from the side dataset.

2. RELATED WORK

In typical unconditional image generation, the learner is provided with a large collection of images and is asked to model the underlying distribution of those images as well as generate novel samples from this distribution [1, 2, 3, 4]. At the other end of the scale, recent methods model the internal distribution of patches of a single image, thus enabling the generation of novel samples that depict the same internal statistics as the single image [5, 6, 7]. In this work, we are interested in generating images from the joint patch distribution of a small number of samples $N > 1$. This is different from recent few-shot generation setting, which attempt to model the external distribution of a new domain given only a few images at training/fine-tuning (e.g. faces to emoji, photo to sketch, etc.) [8, 9, 10, 11]. Instead, we are interested in modeling the internal joint patch distribution of all N images and generating novel samples that capture these patches. As shown in Sec. 4, other models are either unable to learn with such small number of samples (mode collapse), or are unable to generate realistic images, as they place patches from different images in an unrealistic manner.

3. METHOD

We employ $T + 1$ scales, where 0 is the coarsest scale, and T is the finest scale. The role of the first scale is to generate structural diversity. The generator at scale $t = 1, \ldots, T$ is trained to provide a residual signal. Thus, each generator progressively adds detail to the upscaled version obtained from the previous generator.

The coarsest scale 0 consists of a fully convolutional encoder E and a decoder Dec. E encodes image patches to a fixed set of codes from a learned, discrete codebook $\mathbf{Z} =$

$\{z_k\}_{k=1}^{K} \subset R^{n_z}$. K is the number of codebook entries and n_z is the dimensionality of codes.

For a given image $x_0 \in \mathbb{R}^{H \times W \times 3}$, the generation at this scale is purely conditioned:

$$\hat{x}_0 = G_0(x_0) = Dec(Z(E(x_0))) \tag{1}$$

E maps x_0 into a latent space code of shape $z \in \mathbb{R}^{h \times w \times n_z}$. z can be viewed as the encoding of $h \times w$ patches of dimension n_z. Each such patch encoding, $z^{ij} \in \mathbb{R}^{n_z}$, is quantized using Z, the quantization operator, onto its closest codebook entry z_k, creating a quantized version of z, z_q. Next, Dec generates $\hat{x}_0 \in \mathbb{R}^{H \times W \times 3}$ conditioned on z_q. For ease of notation we denote $Dec(Z(E(\cdot)))$ as $G_0(\cdot)$.

In order to learn a context-rich vocabulary of the joint patch distribution of the training set, we apply adversarial loss (Eq. 2) and reconstruction loss (Eq. 3) to the training samples:

$$\begin{aligned}\mathcal{L}_{Adv_0}\{G_0, D_0\}(x_0) = \min_{G_0} \max_{D_0} (\mathbf{E}[D_0(x_0)] - \mathbf{E}[D_0(\hat{x}_0)] \\ - \lambda_{gp} \mathbf{E}[(\|\nabla_{\overline{x}_0} D_0(\overline{x}_0) - 1\|_2)^2])\end{aligned} \tag{2}$$

The adversarial loss is given by the WGAN-GP [12] loss, where $\mathbf{E}$ is the mean over D_0's output, $\overline{x}_0 = \varepsilon x_0 + (1-\varepsilon)\hat{x}_0$, for ε sampled uniformly between 0 and 1, and λ_{gp} is the gradient penalty weight. The VQ loss term is given by:

$$\begin{aligned}\mathcal{L}_{VQ}\{G_0\}(x_0) = \|x_0 - \hat{x}_0\|_2^2 \\ + \|sg[E(x_0)] - z_q\|_2^2 + \beta \|E(x_0) - sg(z_q)\|_2^2\end{aligned} \tag{3}$$

where sg denotes the stop-gradient operation, which passes zero gradient during backpropagation. The first term is the reconstruction loss, the second term is the codebook loss, and the third term is the commitment loss, with β set to 0.25, as in VQ-VAE [13].

In order to generate images of a diverse structure, we use a side-dataset, which is separate from the training set. This is used as a source of randomness for the generator and later as a training set for the auto-regressive generator. To keep the structure of an external image, we add $\mathcal{L}_{SSIM}\{G_0\}(s_0, \hat{s}_0)$ [14] loss between the generated image and the external image. Since we want to keep only the structure, while generating content only from the joint patch distribution of the training set, we apply adversarial loss only for the generator, i.e. the discriminator does not see any image from the side-dataset as real input.

Denote by $s_0 \in \mathbb{R}^{H \times W \times 3}$ an image from the side-dataset in the coarsest scale. The loss is as follows:

$$\mathcal{L}_{Adv-ref_0}\{G_0\}(s_0) = -\mathbf{E}[D_0(\hat{s}_0)] \tag{4}$$

where $\hat{s}_0 = G_0(s_0)$ and $\mathbf{E}$ is the mean over the output of D_0.

We also wish to encourage the model to learn a realistic mapping between patches, e.g., prevent it from mapping water as sky. To enforce this, we concatenate pixel coordinates to the encoded vector before the vector quantization layer. In this manner, each codebook entry also has positional encoding embedded into it. The positional encoding pixel coordinates are defined as follows for each pair of pixel coordinates (i,j): $i' = \frac{2i}{W-1} - 1$ and $j' = \frac{2j}{H-1} - 1$, uniformly mapped to the range $[-1, 1]$. Since the dimensionality of the code vector n_z is usually much larger than 2 coordinates (i', j'), we concatenate them λ_{pos} times, where λ_{pos} is a hyperparameter.

In addition, since the coarsest scale sets the structure for the rest of the scales, we want to make sure that the structure is semantically continuous, i.e. that adjacent patches, of the same semantic object, will be from the patch distribution of the same image (e.g. when generating sky, use the same distribution of a specific image). To enforce this, we apply spatial continuity loss, originally introduced in [15], to the encoder's output (for both x_0 and s_0):

$$\mathcal{L}_{Continuity}\{E\}(x_0, s_0) = \mathcal{L}_{Con}(E(x_0)) + \mathcal{L}_{Con}(E(s_0))$$

where the spatial continuity loss $\mathcal{L}_{Con}$ is defined as follows:

$$\mathcal{L}_{Con}(m) = \sum_{i=1}^{W-1} \sum_{j=1}^{H-1} \|m_{i+1,j} - m_{i,j}\|_1 + \|m_{i,j+1} - m_{i,j}\|_1$$

where W and H represent the width and height of an input image, while $m_{i,j}$ represents the pixel value at (i, j) in the encoded output $m = E(\cdot)$.

Following the first scale, in scales 1 to T, our method employs a patch-GAN [16] for each scale, using a generator G_t and discriminator D_t. As stated, G_t is trained in a residual manner, learning to add details to samples from the previous scale. For $t > 0$, let $\hat{x}_{t-1}$ be the output of the previous scale, $\uparrow \hat{x}_{t-1}$ be the result of upsampling $\hat{x}_{t-1}$ to the scale of level t, and x_t the real input image for scale t. We define $\hat{x}_t$ to be:

$$\hat{x}_t = \uparrow \hat{x}_{t-1} + G_t(\uparrow \hat{x}_{t-1}) \tag{5}$$

D_t produces a single-channel activation map of the same dimension as its input, indicating whether each patch of the input is real or fake, based on the effective receptive field r.

When using a patch-GAN, we follow a similar procedure to SinGAN [5] and HP-VAE-GAN [6], using a fully convolutional generator and a discriminator of a fixed effective receptive field r, while varying the resolution of x at each scale. In scales 1 to T, the receptive fields become smaller, and the top-level generators introduce fine textural details. At these scales, we wish to encourage quality over diversity, which patch-GAN [16] does well.

When training each scale $t > 0$, only G_t and D_t are trained, while $G_0, \ldots, G_{t-1}$ are frozen. The loss used is:

$$\begin{aligned}\min_{G_t} \max_{D_t} \mathcal{L}_t\{G_t, D_t\} = \mathcal{L}_{Adv_t}\{G_t, D_t\} + \mathcal{L}_{Adv-ref_t}\{G_t\} \\ + \mathcal{L}_{SSIM}\{G_t\} + \mathcal{L}_{Reconstruction}\{G_t\}\end{aligned} \tag{6}$$

such that $\mathcal{L}_{Adv_t}\{G_t, D_t\}$, $\mathcal{L}_{Adv-ref_t}\{G_t\}$ and $\mathcal{L}_{SSIM}\{G_t\}$ are defined the same as for scale 0, but with G_t (resp. D_t)

instead of G_0 (resp. D_0). $\mathcal{L}_{Reconstruction}$ is defined as $\|x_t - \hat{x}_t\|_2^2$.

With E and Z available, we can now represent the side-dataset images in terms of the codebook indices of their encodings. For a given image $s \in \mathbb{R}^{H \times W \times 3}$, we consider $c = Z(E(s)) \in \{0, \ldots, K-1\}^{H*W}$. Image generation can then be formulated as an auto-regressive prediction: Given indices $c_{<i}$, a PixelCNN model learns to predict the distribution of the next location, i.e. $\mathcal{P}(c_i|c_{<i})$. The training procedure directly maximizes the log-likelihood with respect to this autoregressive processing of the images s in the side dataset.

Inference Once we have the PixelCNN trained, we can generate c via ancestral sampling, and then use $Dec, G_1, \ldots, G_T$ to unconditionally generate novel images from the joint patch distribution of the training images.

Using FewGAN for image manipulation tasks, such as editing and harmonization, is straightforward by applying $G_0, \ldots, G_T$ on the edited input. As for multi-modal inpainting, the multi-modality is derived from the PixelCNN model in the following manner: (1) mask the corresponding pixels of the edited input in the discrete latent space, i.e. after applying E and Z. (2) PixelCNN fills the occluded section. (3) apply $Dec, G_1, \ldots, G_T$ on the output of PixelCNN.

4. RESULTS

To the best of our knowledge, our work is the first that enables the generation of novel images from the joint patch distribution of a small number of training samples. We have tested a number of methods that require a large collection of images and they resulted in a mode collapse [3, 4, 2, 13]. Methods dedicated to handling relatively small datasets [17, 18, 19] also result in mode collapse when using only few images.

As for single-image generation methods, SinGAN's [5] extension to multiple images was not successful. On the other hand, we were able to extend GPNN [7], the nearest neighbor version of SinGAN [5], to generate images conditionally, by using patches from multiple images. Since this method lacks the ability of unconditional generation, we only show its result in the qualitative section. We do not include it in the quantitative evaluation, since it is based on unconditional evaluation only. We were able to successfully extend HP-VAE-GAN [6] to work with multiple images; therefore, it will be used as our primary baseline. The extension was straightforward by simply passing multiple images into the VAE [20] in the coarsest scale, thus enabling both conditional and unconditional generation.

Since the number of images N is quite small, we have also tried to concatenate all the training images into a single image, and to train a single-image generative model on that image. We used SinGAN [5] as the single-image generative model, and call this baseline Concat-SinGAN. The results of this method are relatively poor and unrealistic, so we only show the quantitative results of this baseline.

Method	KID↓	FID↓	Realism↑	Diversity↑
Our FewGAN	0.036	164	3.6	0.46
HP-VAE-GAN	0.070	208	1.7	0.42
Concat-SinGAN	0.38	359	1.1	0.33

Table 1: Quantitative comparison over datasets of 2,...,10 landscape images.

To evaluate our method quantitatively, we evaluate the realism and diversity of the generated samples. For realism, we use the KID [21] measure. FID [22] is also reported, despite being shown to be unreliable for small datasets, due to its bias towards the dataset size N [21, 17, 23]. To further evaluate realism, we conducted a user study. Our study involved 25 users and 15 images generated by our method, the primary baseline (HP-VAE-GAN [6]) and Concat-SinGAN. For each image, users were asked to rank how real the generated image looks on a scale of 1 to 5. For diversity, we use the measure introduced in SinGAN [5], which computes the average standard deviation over all pixel values along the channel axis of 200 generated images.

Tab. 1 reports these results. FewGAN is clearly superior in every qualitative metric (KID [21], FID [22]). The margin is even larger according to the results of the user survey (realism measure). We also generate more diverse images, as can be seen by the diversity measure.

Qualitative results can be seen in Fig. 1 for multiple landscape datasets ranging between 3 to 6 training images only. For each of these datasets, we compare unconditional generation against HP-VAE-GAN [6]. For conditional generation, we also compare against GPNN [7]. In addition, we demonstrate the results of our model for a variety of image manipulation tasks: editing, inpainting and harmonization.

5. CONCLUSIONS

We introduced FewGAN, the first method that enables unsupervised generation of coherent images from the joint patch distribution of a small number of $N > 1$ samples. We demonstrated its ability to go beyond textures and generate diverse realistic samples for natural complex images. Since this method is fully unsupervised, it has some limitations: (1) It may create a semantically wrong mapping between patches if they are nearest neighbors (e.g. map water to sky). (2) It may select different patches from different images to represent a single semantic entity (e.g. use the sky in two images to represent "sky"). Although we addressed these issues in our method - by adding positional encoding and continuity loss - they may still occur, but are less likely than with other methods we compared it with, as can be seen from our user survey and the qualitative comparison. FewGAN can provide a very powerful tool for a wide range of image manipulation tasks, as demonstrated in the qualitative evaluation.

Figure 1: qualitative evaluation and baseline comparison. Top figures: 6 images dataset comparison and evaluation of image manipulation tasks: editing, inpainting and harmonization. Bottom figures: additional comparison with other datasets against our primary baseline.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant ERC CoG 725974).

6. REFERENCES

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial networks," 2014.

[2] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila, "Analyzing and improving the image quality of stylegan," 2020.

[3] Patrick Esser, Robin Rombach, and Björn Ommer, "Taming transformers for high-resolution image synthesis," 2021.

[4] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei A. Efros, and Richard Zhang, "Swapping autoencoder for deep image manipulation," 2020.

[5] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli, "Singan: Learning a generative model from a single natural image," 2019.

[6] Shir Gur, Sagie Benaim, and Lior Wolf, "Hierarchical patch vae-gan: Generating diverse videos from a single sample," 2020.

[7] Niv Granot, Ben Feinstein, Assaf Shocher, Shai Bagon, and Michal Irani, "Drop the gan: In defense of patches nearest neighbors as single image generative models," 2021.

[8] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or, "Stylegan-nada: Clip-guided domain adaptation of image generators," 2021.

[9] Yijun Li, Richard Zhang, Jingwan Lu, and Eli Shechtman, "Few-shot image generation with elastic weight consolidation," 2020.

[10] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han, "Differentiable augmentation for data-efficient gan training," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7559–7570, 2020.

[11] Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal, "Towards faster and stabilized gan training for high-fidelity few-shot image synthesis," in *International Conference on Learning Representations*, 2020.

[12] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville, "Improved training of wasserstein gans," 2017.

[13] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu, "Neural discrete representation learning," 2018.

[14] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[15] Wonjik Kim, Asako Kanezaki, and Masayuki Tanaka, "Unsupervised learning of image segmentation based on differentiable feature clustering," *IEEE Transactions on Image Processing*, vol. 29, pp. 8055–8068, 2020.

[16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros, "Image-to-image translation with conditional adversarial networks," 2018.

[17] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila, "Training generative adversarial networks with limited data," 2020.

[18] Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal, "Towards faster and stabilized gan training for high-fidelity few-shot image synthesis," 2021.

[19] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han, "Differentiable augmentation for data-efficient gan training," 2020.

[20] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," 2014.

[21] Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton, "Demystifying mmd gans," 2021.

[22] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," 2018.

[23] Min Jin Chong and David Forsyth, "Effectively unbiased fid and inception score and where to find them," 2020.