Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2022) 30: 1741 – 1757
© TÜBİTAK
doi:10.55730/1300-0632.3902

Content loss and conditional space relationship in conditional generative adversarial networks

Enes EKEN[*]
Electrical and Electronics Engineering, Faculty of Engineering, Aksaray University, Aksaray, Turkey

Received: 27.08.2021 • **Accepted/Published Online:** 27.05.2022 • **Final Version:** 22.07.2022

Abstract: In the machine learning community, generative models, especially generative adversarial networks (GANs) continue to be an attractive yet challenging research topic. Right after the invention of GAN, many GAN models have been proposed by the researchers with the same goal: creating better images. The first and foremost feature that a GAN model should have is that creating realistic images that cannot be distinguished from genuine ones. A large portion of the GAN models proposed to this end have a common approach which can be defined as factoring the image generation process into multiple states for decomposing the difficult task into several more manageable sub tasks. This can be realized by using sequential conditional/unconditional generators. Although images generated by sequential generators experimentally prove the effectiveness of this approach, visually inspecting the generated images are far away of being objective and it is not yet quantitatively showed in an objective manner.

In this paper, we quantitatively show the effectiveness of shrinking the conditional space by using the sequential generators instead of utilizing single but large generator. At the light of the content loss we demonstrate that in sequential designs, each generator helps to shrink the conditional space, and therefore reduces the loss and the uncertainties at the generated images. In order to quantitatively validate this approach, we tried different combinations of connecting generators sequentially and/or increasing the capacity of generators and using single or multiple discriminators under four different scenarios applied to image-to-image translation tasks. Scenario-1 uses the conventional pix2pix GAN model which serves as the based line model for the rest of the scenarios. In Scenario-2, we utilized two generators connected sequentially. Each generator is identical to the one used in Scenario-1. Another possibility is just doubling the size of a single generator which is evaluated in the Scenario-3. In the last scenario, we used two different discriminators in order to train two sequentially connected generators. Our quantitative results support that simply increasing the capacity of one generator, instead of using sequential generators, does not help a lot to reduce the content loss which is used in addition to adversarial loss and hence does not create better images.

Key words: Generative adversarial networks, conditional space, content loss, sequential generators, image-to-image translation

1. Introduction

In the last decade we have witnessed that many challenging image processing problems, previously required domain specific and hand-crafted solutions ranging from image classification [1] to object detection [2], have been solved successfully with machine learning methods. Although the problems are very different in the image processing field, proposed solutions, in general, have the same backbone, namely, deep neural networks (DNNs). Especially in certain discriminative tasks, deep convolutional neural networks (CNNs) [3] were utilized

[*]Correspondence: eneseken@aksaray.edu.tr

This work is licensed under a Creative Commons Attribution 4.0 International License.

as hierarchical feature extractors and with this functionality CNNs have achieved impressive results.

In addition to discriminative tasks, generative models also have made unprecedented success stories in difficult, ill-posed problems such as composing music [4] and painting art [5]. Among the different generative models such as variational auto encoder (VAE) [6], auto regressive model [7]; generative adversarial networks (GANs) [8] have drawn attentions both from academia and industry due to its huge application capacity spanning from reinforcement learning [9] to image-to-image translation [10]. Despite the fact that other generative models also have their own positive and negative sides e.g., training stability, generating sharp/bluer images, GANs became prominent in this area with its features such as not requiring Monte Carlo simulations, generating sharp images, whereas it is notoriously difficult to train them [11].

A GAN framework is comprised of two models, namely, a generative model G and a discriminative model D. Both of these two models can be realized as multilayer perceptrons and can be trained with backpropagation from end-to-end. These models race with each other in an iterative game. In this game, the primary purpose of the generator is to synthesize fake images resembling real ones, whereas the discriminative model tries to distinguish real images from the generated fake images. The generative model receives a random input vector $\boldsymbol{z}$ in $\mathbb{R}^d$ sampled from $p_{\mathbf{z}}(\cdot)$ and maps it to fake image $\hat{\mathbf{x}} = \mathrm{G}(\boldsymbol{z};\theta^G)$, where θ^G is the generator's parameters. $\boldsymbol{z}$ can have different distributions such as multivariate normal distribution or a uniform distribution with a support of $[-1\ \ 1]^d$. On the other hand, the discriminator takes as input real images $\mathbf{x}$ sampled from $p_{\mathbf{x}}(\cdot)$ or fake images $\hat{\mathbf{x}}$ sampled from $p_{\mathbf{g}}(\cdot)$ (here, we denote the distribution of $\mathrm{G}(\boldsymbol{z};\theta^G)$ as $p_{\mathbf{g}}(\cdot)$) and assigns probability $\mathrm{D}(\mathbf{x},\theta^D)$ or $\mathrm{D}(\mathrm{G}(\boldsymbol{z};\theta^G),\theta^D)$, where θ^D is the discriminator's parameters. Ideally speaking, $\mathrm{D}(\mathbf{x},\theta^D)$ should be 1, and $\mathrm{D}(\mathrm{G}(\boldsymbol{z};\theta^G),\theta^D)$ should be 0. The seminal work of Goodfellow et al. [8] proves that given enough capacity to the generator and discriminator and sufficient training iterations, the distribution of generator, $p_{\mathbf{g}}(\cdot)$, converges to real image distribution, $p_{\mathbf{x}}(\cdot)$, which means that it is no more possible for discriminator to distinguish fake images from the real ones.

Right after the invention of the GAN, many different GAN models have been proposed by the researchers in different application areas ranging from image super resolution [12] to image-to-image translation [10]. Despite the fact that there are a plethora of GAN models in the literature [13], in general they have two main purposes: synthesizing diverse and realistic data. Some of the proposed models, in accordance with these objectives, have a similar approach which can be described as partitioning a difficult task into smaller and more manageable subtasks.

Although each GAN model has its own unique architecture, this approach is commonly used in some form by different GAN models. For example Denton et al. [14] proposed a LAPGAN model to generate high quality images which employs a multiscale approach, connecting generators sequentially in a coarse-fine fashion, and gradually generating higher resolution images. Wang and Gupta [15] proposed a S^2-GAN which employs two generators, one for generating surface normals and another for generating images conditioned on surface normals. Huang et al. [16] proposed SGAN which uses a top-down stack of GAN in order to utilize hierarchical representations of a bottom-up discriminator network. Zhang et al. [17] came up with StackGAN which decomposes image synthesis into two stages: Low-resolution images are first generated by Stage-I GAN and top of it they stack Stage-II GAN to generate high resolution images conditioned on Stage-I GAN. Although these aforementioned GAN models are very different in terms of architectural style, the approach of "breaking the image generation process into sequential and more manageable subtasks" is showing similarity. In fact the effectiveness of this approach is experimentally showed based on the high quality images generated by these models. However, images are not very informative to analyze and evaluate the quantitative effects of using sequential generators or using single but large generator.

In this paper, we filled this gap and quantitatively analyzed how a GAN model behaves in certain design changes. Possible architectural choices could be using sequential generators, preferring single but bigger generator, or utilizing multiple discriminators.

In the sequential case, first generator shrinks the possible conditional space, but still requires sufficient exploration for the optimal solution. Moreover, the second generator searches the best possible solution based on the conditional space. Instead of using two sequential generators, utilizing single but larger generator is another option. However, simply using bigger generators do not guarantee synthesizing more realistic images and thus researchers generally refrain from using simply bigger generators. We also take into account using single or multiple discriminators in our extensive quantitative analysis at the light of the content loss. Different than the classification models where the loss function e.g., MSE is being used for monitoring the training of the model, training a GAN model is like playing a zero sum min-max game where each player wants to demolish opponent's accomplishment. Therefore, adversarial loss is not very informative to monitor the training of the GAN. This well-known phenomenon is reported by several researchers [18][19]. Although adversarial loss is not very informative to investigate the effect of the shrinking the conditional space on the generated image, the content loss sheds light on it. It is worth mentioning that, although the topic is generic, to be able to use aforementioned scenarios, the content loss is crucial, which requires conditioning the GAN model and therefore, currently it is only applicable to the image-to-image translation GANs such as pix2pix [10] and CycleGAN [20]. Our code is available at: https://github.com/enexs.

The rest of the paper is organized as follows: In Section II we explained different GAN models that share conceptual similarities. In Section III we demonstrate our quantitative analysis which relates the content loss and the conditional space. And we conclude our work in Section IV.

2. Related work

In theory, giving enough capacity for generator and discriminator networks along with sufficient training iteration, GAN [8] training procedure leads a generator distribution that perfectly matches the real data distribution. On the other hand, practically speaking, training GANs is notoriously difficult. In contrast to DNNs trained for discriminative tasks with stochastic gradient descents, in GANs, both the generator model and the discriminator model are trained simultaneously in min-max game with value function $V(G,D)$:

$$\min_{G}\max_{D} V(G,D) = \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{z\sim p_{\mathbf{z}}(\mathbf{z})}[\log(1-D(G(\mathbf{z})))]. \tag{1}$$

In practice Equation 1 is solved in an iterative manner following two gradient update steps:

$$\begin{aligned}\theta^D_{t+1} &= \theta^D_t + \lambda_t \nabla_{\theta^D} V(G_t, D_t)\\ \theta^G_{t+1} &= \theta^G_t - \lambda_t \nabla_{\theta^G} V(G_t, D_{t+1}),\end{aligned} \tag{2}$$

and this can cause to nonconvergence to Nash equilibrium [21]. Among the many other challenging issues in training GAN such as, "mode collapse" which can be defined as generating same output although receiving different inputs [22], evaluating different GAN models [23, 24]; increasing image fidelity is another main branch of GAN researches.

In order to increase image fidelity and generate photo realistic images Denton et al. [14] proposed LAPGAN. Main philosophy behind the LAPGAN is that partitioning a difficult task into multiple subtasks which are more manageable compared to original one. In the framework of the GAN, this idea was realized by using sequential convolutional neural network based generators each of which at different scale such as $8 \rightarrow 16 \rightarrow 32 \rightarrow 64 \rightarrow 96$. Each generator receives input from previous one and generates output image conditioned on the input in a coarse-to-fine fashion. The cascade generators and discriminators in this Laplacian pyramid are trained separately first and then fine tuned together. With this approach LAPGAN succeeded to generate higher quality images at that time compared to its counterparts.

Wang and Gupta [15] suggested that images are actually combination of two separate parts, namely, the structure and the style, and these parts should be generated sequentially. While the structure of an image defines the underlying 3D geometry of the scene, the style determines the texture of the image. As parallel to this process, they proposed S^2-GAN which factors the image generation process of GAN into two generative processes. As a first step, structure-GAN receives random input vector $\boldsymbol{z}$ as normal, but instead of the whole image, it is trained for only synthesizing the underlying 3D structure for the scene. Subsequently, the Style-GAN takes this 3D structure of the image as input and generates the output image conditioned on the input image. Since this two-step procedure simplifies the over all image generation process, S^2-GAN was able to generate more realistic high resolution images.

Huang et al. [16] proposed a novel GAN architecture, stacked generative adversarial networks (SGAN), which bridged the gap between bottom-up neural networks trained for discriminative tasks and top-down generative models for leveraging the representation power of a pretrained DNNs. In this architecture each GAN in the stack is trained to invert the hierarchical representations of bottom-up discriminative network for the corresponding layer of the DNN. Instead of lower resolution images, intermediate generators synthesize lower level representations conditioned on higher level representations. Similarly, each discriminator network in the stack is responsible for differentiating real and fake representations instead of real and fake images and trained for this purpose. Notwithstanding the novelty of the architectural design, partitioning the image synthesizing task into multiple steps is efficiently utilized by SGAN, too.

Generating high fidelity images conditioned on given text description is another popular application of GANs. In this domain, Zhang et al. [17] brought forward stacked generative adversarial networks (StackGAN) which is capable of generating 256256 photo-realistic images conditioned on text description. In order to accomplish this, StackGAN decomposes image generation process into two subtasks as Stage-I and Stage-II. The Stage-I GAN draws the general shape and the color of the object at 6464 resolution based on the given text description. Conditioned on the output of the Stage-I GAN, Stage-II GAN adds more compelling details to the image and generates 256256 photo-realistic images. With their unique architecture, Zhang et al. succeeded to generate 256256 images which indicates the benefits of sequential GAN designs where each GAN is responsible of a part of big picture.

Since the very beginning of invention of the GAN, it has been studied by a large group of researchers both in industry and academia. The technique of using sequential generators has been previously proposed and used in different architectures in different forms. Nonetheless earlier papers have focused on generating better images in terms of image size and image fidelity.

Although these works resemble in spirit what we do, technically our paper is very different. Our primary contribution is not proposing another member of GAN family but investigating the effect of shrinking the conditional space on the generated image quality. In order to evaluate this in an objective manner, we considered different possible combinations. We first evaluate the effect of simply increasing the capacity of a single generator in a GAN. Following this, we considered another option, using two sequential generators instead of increasing the size of one generator. On this architecture we also take into account the effect of using single or multiple discriminators. In order to evaluate different options and compare with each other quantitatively, we used content loss which can be defined as the distance between generated image and ground truth output, instead of presenting the generated images which are far away of being objective. For this setup we used pix2pix [10] GAN framework as a benchmark which has shown high quality results in the supervised image-to-image translation setting.

3. Effect of conditional space on the content loss

As a theoretical definition, GAN is a directed graphical model, which receives random input vector $\boldsymbol{z}$ sampled from $p_{\mathbf{z}}(\cdot)$ and returns $\hat{\mathbf{x}}$. In this setting, the generator G implicitly defines a conditional density model $p_{\mathbf{g}|\mathbf{z}}(\hat{\mathbf{x}} \mid \mathbf{z})$ which provides us a way to reason about $\hat{\mathbf{x}}$ based on the partial information $\mathbf{z}$, the corresponding conditional space. Obviously inflating or deflating of the conditional space directly affects the certainties of the possible outcomes. One possible way of shrinking the conditional space could be using generated image $\hat{\mathbf{x}}$, as a new conditional space. This can be easily realized as using sequential generators, in which output of first generator goes into the second generator as the input.

Let us assume that we have 'n' number of identical GANs, trained on the same data distribution $p_{\mathbf{x}}(\cdot)$ with the same architecture, except first generator receives $\boldsymbol{z}$ in $\mathbb{R}^d$ and produces $\hat{\mathbf{x}}$ in $\mathbb{R}^{w\text{x}h}$, while subsequent generators receive and produce image in the same space, $\mathbb{R}^{w\text{x}h}$. The approximate joint probability distribution of this directed graph will be;

$$p_{\mathbf{z},\mathbf{g}_1,\mathbf{g}_2,..,\mathbf{g}_n}(z,x_1,x_2,..,x_n) = p_{\mathbf{z}}(z)p_{\mathbf{g}_1|\mathbf{z}}(x_1 \mid z)p_{\mathbf{g}_2|\mathbf{z},\mathbf{g}_1}(x_2 \mid z,x_1)...p_{\mathbf{g}_n|\mathbf{z},\mathbf{g}_1,..\mathbf{g}_{n-1}}(x_n \mid z,x_1,..x_{n-1}). \tag{3}$$

In this graphical structure, output of any generator is conditionally independent of the previous outputs, given the input of that generator and hence can be simplified as;

$$\begin{gathered}\mathbf{g}_{i+1} \perp\!\!\!\perp \mathbf{g}_{i-1}, \mathbf{g}_{i-2}, ...\mathbf{g}_0 \mid \mathbf{g}_i, i = 1, ...n-1, \\ p_{\mathbf{g}_0,\mathbf{g}_1,\mathbf{g}_2,..,\mathbf{g}_n}(x_0,x_1,x_2,..,x_n) = p_{\mathbf{g}_0}(x_0)\prod_{i=0}^{n-1} p_{\mathbf{g}_{i+1}|\mathbf{g}_i}(x_{i+1} \mid x_i).\end{gathered} \tag{4}$$

Here, for the sake of simplicity we demonstrate random variable $\boldsymbol{z}$ as $\mathbf{g}_0$ and its value as x_0. Similar to Markov chain assumption, future state is only dependent to the value of the current state but not the past. Less formal and more pedagogical demonstration can be seen in Figure 1. As can be seen from the figure, the conditional space of first generator g_0, is random input vector $\boldsymbol{z}$ and as a result of this, its outputs cover more broadly results with low probability. On the other hand, last generator g_2 is conditioned on x_1 and more certain about its outputs.

The same concept can also be explained by using information theory, since each generator helps reducing uncertainty of the output. The joint entropy of the system will be the sum of the conditional entropy as the

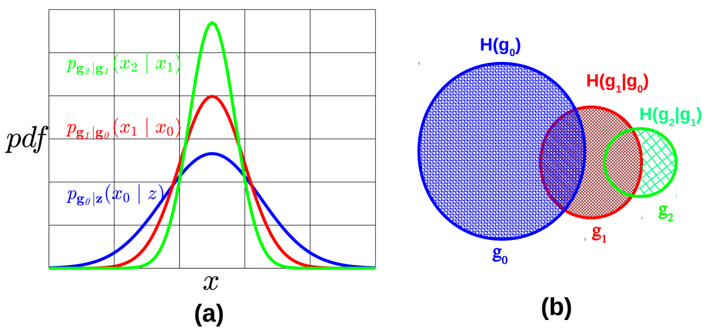

$p_{\mathbf{g}_2|\mathbf{g}_1}(x_2 \mid x_1)$
$p_{\mathbf{g}_1|\mathbf{g}_0}(x_1 \mid x_0)$
$p_{\mathbf{g}_0|\mathbf{z}}(x_0 \mid z)$
pdf
x
(a)
H(g0)
H(g1|g0)
H(g2|g1)
g0
g1
g2
(b)

Figure 1. In sequential GAN design, each generator receives input from previous one and generates output conditioned on the input. (a) Shrinking the conditional space increases the certainty. (b) Total entropy of sequential GANs can be expressed as the sum of the conditional entropies based on the chain rule as stated at Eq. 6

chain rule implies;

$$\begin{aligned} H(\mathbf{g}_0, \mathbf{g}_1, ...\mathbf{g}_n) &= -\sum_{x_0,x_1,..x_n} p_{\mathbf{g}_0,\mathbf{g}_1,..\mathbf{g}_n}(x_0, x_1, ..x_n) \log p_{\mathbf{g}_0,\mathbf{g}_1,..\mathbf{g}_n}(x_0, x_1, ..x_n) \\ &= -\sum_{x_0,x_1,..x_n} p_{\mathbf{g}_0,\mathbf{g}_1,..\mathbf{g}_n}(x_0, x_1, ..x_n) \log \prod_{i=0}^{n} p_{\mathbf{g}_i|\mathbf{g}_{i-1}}(x_i \mid x_{i-1}) \\ &= -\sum_{x_0,x_1,..x_n} \sum_{i=0}^{n} p_{\mathbf{g}_0,\mathbf{g}_1,..\mathbf{g}_n}(x_0, x_1, ..x_n) \log p_{\mathbf{g}_i|\mathbf{g}_{i-1}}(x_i \mid x_{i-1}) \\ &= \sum_{i=0}^{n} H(\mathbf{g}_i \mid \mathbf{g}_{i-1}). \end{aligned} \tag{5}$$

The Venn diagram demonstrated at Figure 1 (b) expresses the total entropy of a GAN system with 3 generators. Using the conditional independence and applying the chain rule of the entropy, we can simply express this system as;

$$H(\mathbf{g}_0, \mathbf{g}_1, \mathbf{g}_2) = H(\mathbf{g}_0) + H(\mathbf{g}_1 \mid \mathbf{g}_0) + H(\mathbf{g}_2 \mid \mathbf{g}_1) \tag{6}$$

which explains the partitioning of one particular problem into more manageable subproblems.

In order to evaluate this approach extensively, we conducted our simulations at the image-to-image translation task which is an attractive application of GAN.

Image-to-image translation

Translating an input image from one domain to another domain is considered as image-to-image translation. Converting an image from day to night, changing a photo from summer to winter [25], or a satellite image to map [10] are some of the prominent applications of image-to-image translation. There are two main methods for this task depending on the format of the input-output image pairs, supervised and unsupervised translations. In the unsupervised translation e.g., CycleGAN [20], and DualGAN [26], the image pairs do not have to be aligned for the training. The input and output images are randomly chosen from input and output domains. On the other hand, in the supervised translation e.g., pix2pix [10], the image pairs should be matched, such as satellite image and its corresponding map are required for training.

Datasets

In our simulations we used Maps [10] dataset which contains 1096 training images of New York City scraped from Google Maps and Facades [10] which contains facade and segmentation of 506 buildings. The images are colorful with 256256 pixels. One specific reason for choosing these datasets among the many others is that the both of them have color codes. During translation of the image, the colors cannot be chosen as random. For example highway should be coded as yellow color and the parks should be light green background. Since there is no an ambiguity about the colors, the content loss will be more important. In our architectural setup we used two NVIDIA GTX 1080Ti GPUs in order to train the proposed networks which lasted about $2-6$ h in Tensorflow framework with AMD Ryzen threadripper 1920X 3.5GHz CPU and 64GB memory.

3.1. Scenario-1: baseline model pix2pix

In order to compare different GAN configurations with each other, we chose pix2pix GAN model proposed by [10] as a baseline model for image-to-image translation task. Different than the vanilla GAN [8], pix2pix is a conditional GAN model, namely, the output of the model should not only be realistic but also should be consistent with the input. General diagram of pix2pix is demonstrated at Figure 2. Although it is possible to employ this model in many problems e.g., synthesizing realistic photo from semantic labels or converting an image from night to day, to be consistent we trained this model on aerial image to map translation using MAP dataset. Since pix2pix is a supervised GAN model, the input(x, aerial image) and output(y, map) images are paired. The generator of pix2pix can be seen in Figure 3. The random input vector $\boldsymbol{z}$ used in vanilla GAN is not utilized here. Instead the randomness is introduced as dropout layers in the model architecture. The objective function of pix2pix is composed of two components. The conditional adversarial loss which can be expressed as:

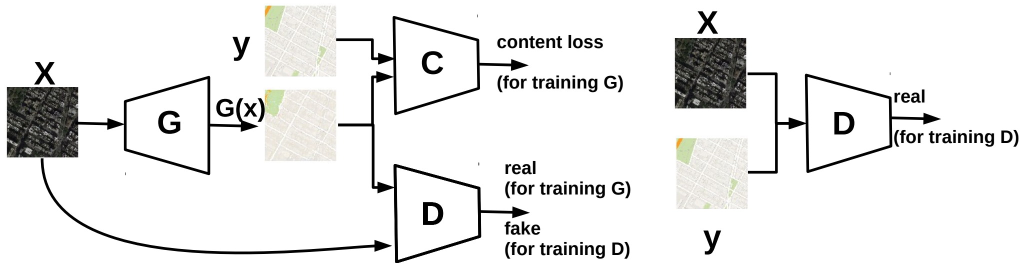


Figure 2. General diagram of pix2pix GAN model which receives x as input and translates it to $G(x)$. This model is trained on paired dataset.

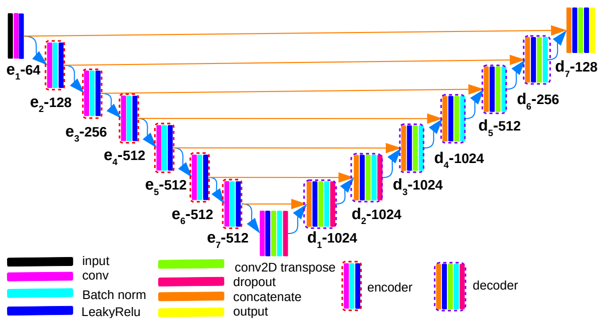


Figure 3. The generator used in the pix2pix model has U-Net architecture which consists of decoder and encoder blocks and skip connections between them. The numbers after the dashed indicates the number of features at each stack.

$$\mathcal{L}_{cGAN}(G,D) = \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_{data}(\mathbf{x},\mathbf{y})}[\log D(\mathbf{x},\mathbf{y})] + \mathbb{E}_{x\sim p_{\mathbf{x}},z\sim p_{\mathbf{z}}(\mathbf{z})}[\log(1-D(\mathbf{x},G(\mathbf{x},\mathbf{z})))], \tag{7}$$

and the content loss is defined as:

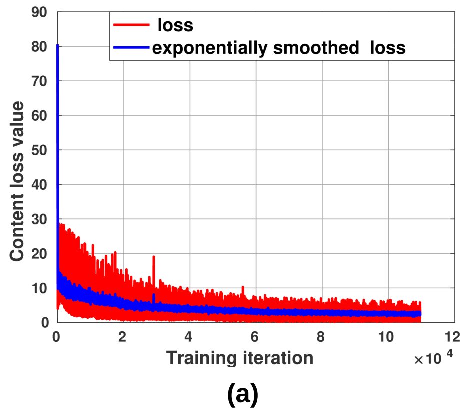
$$\mathcal{L}_{L1}(G) = \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_{data}(\mathbf{x},\mathbf{y}),\mathbf{z}\sim p_{\mathbf{z}}}[\|y - G(x,z))\|_1]. \tag{8}$$

The final objective then will be

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G,D) + \lambda\mathcal{L}_{L1}(G), \tag{9}$$

here λ is hyperparameter and chosen as 100 in our simulations as in the original value.

The change of the content loss during training runs can be seen in Figure 4a. We will compare other GAN designs presented following sections with this result. In order to eliminate the oscillation on the curve and make a fair comparison we also demonstrated the exponentially smoothed version of the content loss. At the rest of the paper, unless otherwise explicitly stated, we used exponentially smoothed versions of the curves. After the training, the prediction $(G(x))$ of the pix2pix model for giving input images (x) along with target images (y) are presented at Figure 4-b.



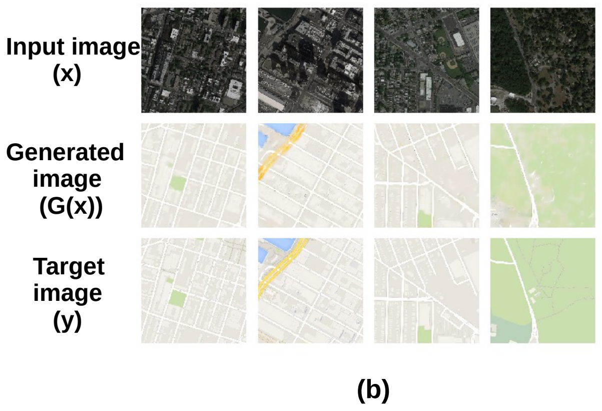


Figure 4. pix2pix model is trained on the Maps dataset which contains paired images (x,y). (a) Changing of the content loss during the trainin runs. (b) Giving the aerial input images, pix2pix translates them to the maps, G(x).

3.2. Scenario-2: two sequential generators

In this scenario we connected two identical generators sequentially as depicted in Figure 5. The architectures of these generators are the same as the one used in the pix2pix model in Scenario-1. While the output of first generator G_1 goes into G_2 and first content loss C_1, the discriminator receives the image from G_2. G_1 does not have its own discriminator and therefore, G_1 receives the gradient from discriminator via G_2. In order to be consistent with original pix2pix model, G_1 is also trained by content loss. The content losses for G_1 and G_2 are defined as

$$\begin{aligned}\mathcal{L}_{L1-G_1}(G_1) &= \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_{data}(\mathbf{x},\mathbf{y}),\mathbf{z_1}\sim p_{\mathbf{z_1}}}[\|y - G_1(x,z_1))\|_1],\\ \mathcal{L}_{L1-G_2}(G_2) &= \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_{data}(\mathbf{x},\mathbf{y}),\mathbf{z_1}\sim p_{\mathbf{z_1}},\mathbf{z_2}\sim p_{\mathbf{z_2}}}[\|y - G_2(G_1(x,z_1),z_2))\|_1].\end{aligned} \tag{10}$$

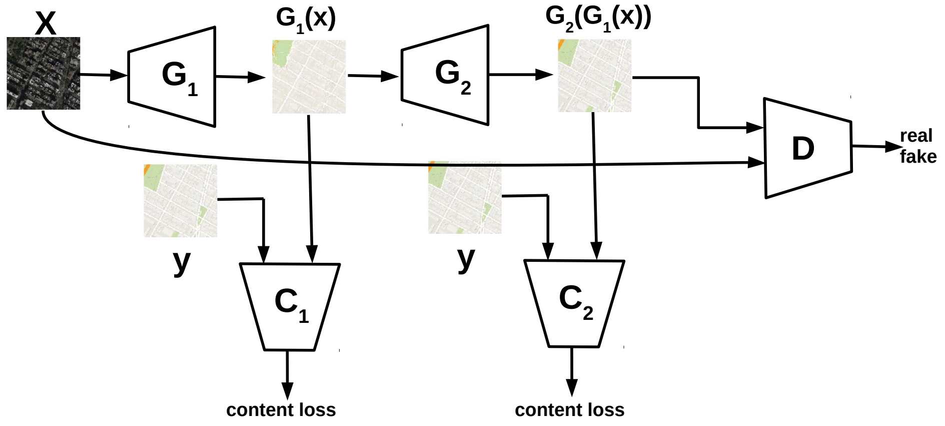


Figure 5. General diagram of GAN model used in Scenario-2 which contains sequential generators and one discriminator connected to the G_2.

With a small modification, the conditional adversarial loss can be expressed as

$$\mathcal{L}_{cGAN}(G_1, G_2, D) = \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_{data}(\mathbf{x},\mathbf{y})}[\log D(\mathbf{x}, \mathbf{y})] + \mathbb{E}_{x\sim p_{\mathbf{x}}, z_2\sim p_{\mathbf{z}_2}(\mathbf{z}_2)}[\log(1 - D(\mathbf{x}, G_2(G_1(\mathbf{x}, \mathbf{z}_1), \mathbf{z}_2)))]. \quad (11)$$

In order to compare the content loss of this sequential model with the previous one, we subtract $\mathcal{L}_{L1-G_2}(G_2)$ (Eq. 10) from $\mathcal{L}_{L1}(G)$ (Eq. 8) and define content loss difference of this model as;

$$\mathcal{L}_{seq} = \mathcal{L}_{L1}(G) - \mathcal{L}_{L1-G_2}(G_2). \quad (12)$$

The graph of $\mathcal{L}_{L1-G_2}(G_2)$ and $\mathcal{L}_{seq}$ can be seen in Figure 6. As depicted in Figure 6-b, between points A and B, $\mathcal{L}_{L1-G_2}(G_2)$ is higher than $\mathcal{L}_{L1}(G)$. This is somewhat expected since sequential model has two generators which means number of free parameters and the number of layers are approximately double of the base model pix2pix. Although the sequential model has higher number of trainable parameters, due to the vanishing gradient phenomena, the cost of it decreases more slowly.

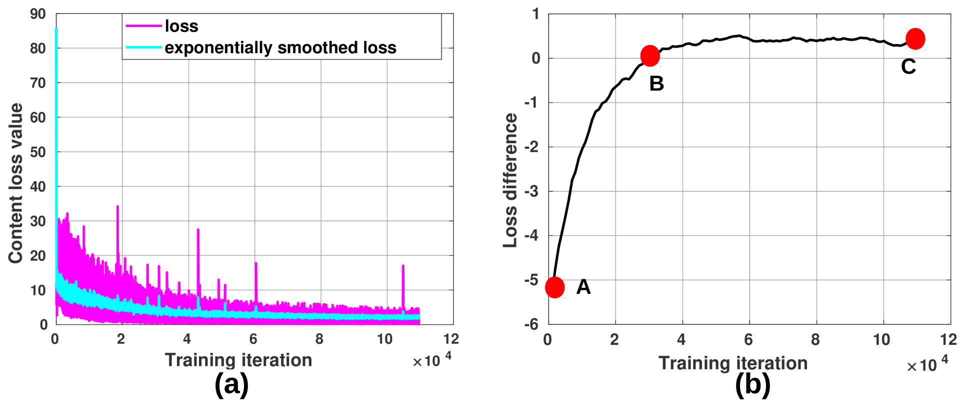


Figure 6. (a) Changing of the content loss during the training of sequential model. (b) Content loss difference between pix2pix and sequential model.

At point B, the costs become equal and after that point, $\mathcal{L}_{L1-G_2}(G_2)$ falls below of $\mathcal{L}_{L1}(G)$ which quantitatively explains that the generated images by sequential model are more close to target images compared to pix2pix. The images generated by this model can be seen Figure 7.

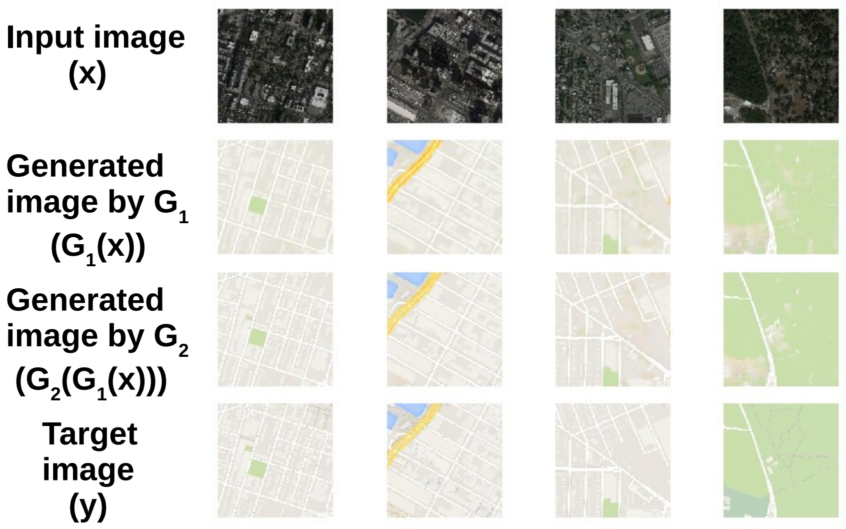

Input image (x)
Generated image by G1 (G1(x))
Generated image by G2 (G2(G1(x)))
Target image (y)

Figure 7. Translated images by sequential generators.

3.3. Scenario-3: one big generator

Another possible option for designing a generator is just increasing its computational capacity instead of sequentially connecting identical generators as presented in previous section. Under Scenario-3 we investigate this design choice. In this scenario the design of the GAN model, so the cost functions, are identical to the ones in Scenario-1, except the generator design. The generator used in Scenario-1, in pix2pix, has a U-Net [27] architecture which is comprised of a series decoder and encoder blocks. In this network architecture there are skip connections between outer-most decoder to outer-most encoder until the bottleneck layer. In the image-to-image translation problems, since the input image and the output image have same underlying structure and differ only in surface appearance, the U-Net architecture helps this information flow without loosing it at the bottleneck layer.

The generator used in Scenario-1 can be seen in Figure 3. In order to increase its capacity and make a fair comparison with Scenario-2, we simply duplicated the layers in the decoder and encoder blocks as can be seen in Figure 8. Original decoders used in pix2pix model have a stack of convolution-batch normalization [28]-activation layers. In order to enlarge this design, we repeated this sequence twice. On the other hand, original decoders consist of concatenate-activation-conv2D transpose-batch normalization-dropout [29] layers (The last 4 decoders do not have dropout layers.). While staying true this order, we repeat the design twice, expect the concatenation layer since there is no need it in the same enlarged decoder design. In this design, we keep the number of feature maps same as presented at Figure 3. We named this big generator as G_B in order to prevent any confusion with the one used in Scenario-1 which is named as G without any subscript.

With this generator design, we keep the number of trainable parameters of this model (97M) as close as possible to the one in Scenario-2 (100M) without damaging the general structure of the U-Net. The difference of the content losses between this scenario and the based scenario is defined as

$$\mathcal{L}_B = \mathcal{L}_{L1}(G) - \mathcal{L}_{L1-G_B}(G_B), \tag{13}$$

where $\mathcal{L}_{L1-G_B}(G_B)$ is defined as

$$\mathcal{L}_{L1-G_B}(G_B) = \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_{data}(\mathbf{x},\mathbf{y}),\mathbf{z}\sim p_\mathbf{z}}[\|y - G_B(x,z))\|_1]. \tag{14}$$

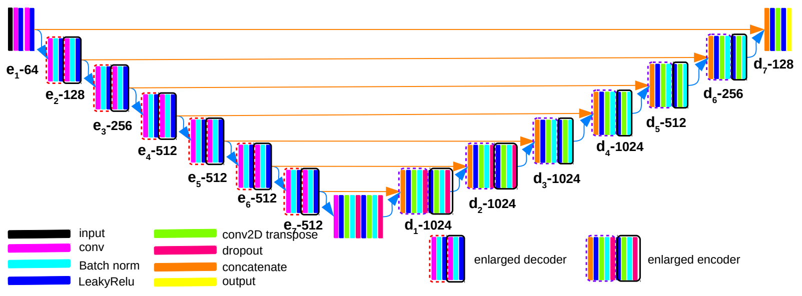


Figure 8. The capacity of the generator, in terms of the trainable parameters, used in Scenario-1 is approximately doubled by appropriately duplicating the layers used in the decoders and encoders.

The graph of $\mathcal{L}_B$ and the generated images can be seen in Figure 9.

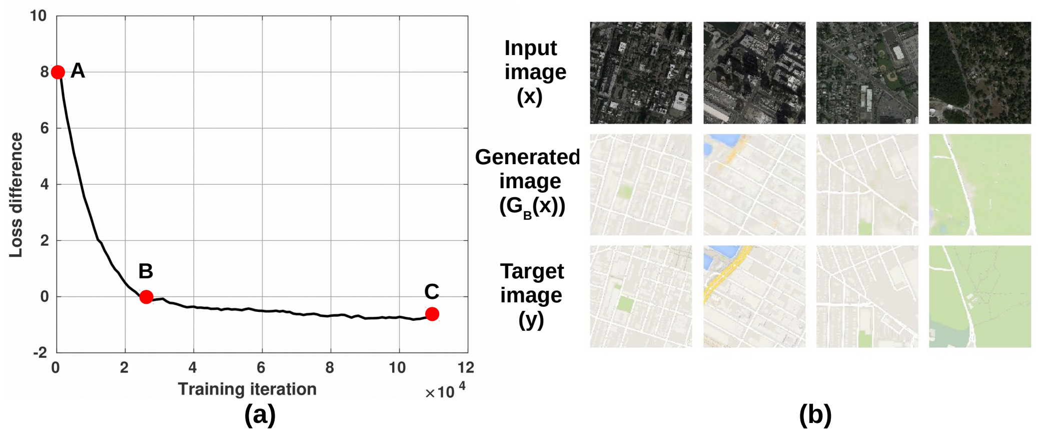


Figure 9. (a) Content loss difference ($\mathcal{L}_B$) between Big model used in Scenario-3 and pix2pix model used as based scenario. (b) Translated images by the Big model.

Due to the skip connections in this enlarged U-Net generator design, the vanishing gradient effect is not being experienced as severely as the sequential model presented at Scenario-2. As our experimental results indicate, using two sequential generators increase the negative effect of vanishing gradient.

Since the computational capacity of this big model is almost doubled by increasing the trainable parameters compared to based model pix2pix, and the gradient vanishing does not severely affect its performance, the cost of $\mathcal{L}_{L1-G_B}(G_B)$ stays below of $\mathcal{L}_{L1}(G)$ between point A and B. At point B, the costs become equal and after that point the situation is reversed and $\mathcal{L}_{L1}(G)$ becomes smaller than $\mathcal{L}_{L1-G_B}(G_B)$ which explains that simply increasing the capacity of the generator does not necessarily mean the generator will synthesize images closer to the ground truth images.

By looking the final values of the costs of three scenarios, we can order them as

$$\mathcal{L}_{L1-G_2}(G_2) < \mathcal{L}_{L1}(G) < \mathcal{L}_{L1-G_B}(G_B). \tag{15}$$

3.4. Scenario-4: two sequential generators with two discriminators

In the last scenario, we strengthen Scenario-2 with individual discriminators. Different than Scenario-2 which uses sequential generators and one discriminator connected to the last generator as depicted in Figure 5, in this case each generator has its own discriminator. As can be seen at Figure 10, output of G_1 not only goes to content loss and G_2 but also feeds D_1. As a direct result of this, G_1 receives feedback from the discriminator in addition to content loss and G_2. This extra feedback mitigates the negative effect of gradient vanishing.

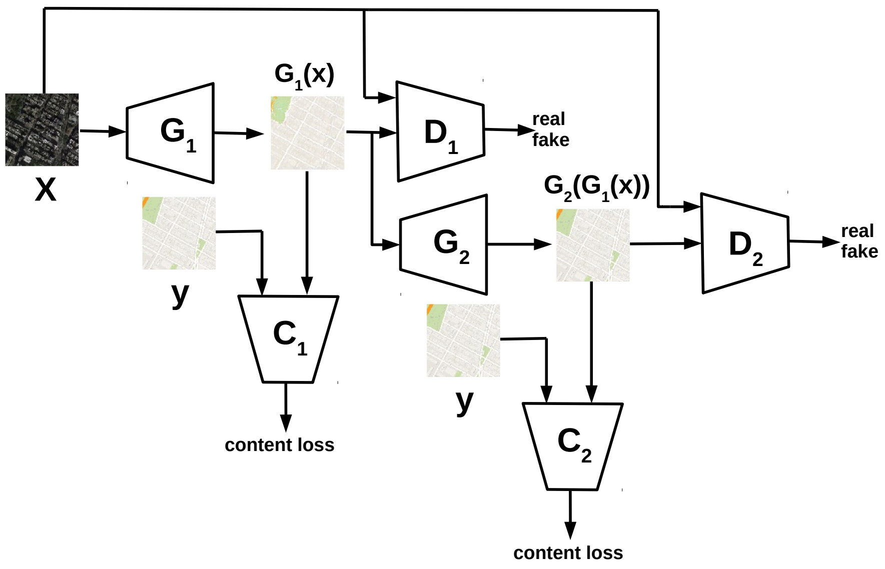


Figure 10. General diagram of the GAN model used in Scenario-4. Diffrent than Scenario-2, here G_1 has its own discriminator D_1.

The content losses of this model is identical to the one in Scenario-2 Equation 10, except, in order to prevent any confusion, we add subscript D, for indicating individual discriminators and defined as

$$\begin{aligned} \mathcal{L}_{L1-G_{1D}}(G_{1D}) &= \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_{data}(\mathbf{x},\mathbf{y}),\mathbf{z_1}\sim p_{\mathbf{z_1}}}[\|y - G_{1D}(x,z_1))\|_1], \\ \mathcal{L}_{L1-G_{2D}}(G_{2D}) &= \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_{data}(\mathbf{x},\mathbf{y}),\mathbf{z_1}\sim p_{\mathbf{z_1}},\mathbf{z_2}\sim p_{\mathbf{z_2}}}[\|y - G_{2D}(G_{1D}(x,z_1),z_2))\|_1]. \end{aligned} \tag{16}$$

The conditional adversarial loss now has two discriminators to train and can expressed as

$$\begin{aligned} \mathcal{L}_{cGAN}(G_{1D},G_{2D},D_{1D},D_{2D}) = \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_{data}(\mathbf{x},\mathbf{y})}[\log D_{1D}(\mathbf{x},\mathbf{y})] + \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_{data}(\mathbf{x},\mathbf{y})}[\log D_{2D}(\mathbf{x},\mathbf{y})] + \\ \mathbb{E}_{x\sim p_{\mathbf{x}},z_1\sim p_{\mathbf{z_1}}(\mathbf{z_1})}[\log(1 - D_{1D}(\mathbf{x},(G_1(\mathbf{x},\mathbf{z_1}))))] + \mathbb{E}_{x\sim p_{\mathbf{x}},z_2\sim p_{\mathbf{z_2}}(\mathbf{z_2})}[\log(1 - D(\mathbf{x},G_2(G_1(\mathbf{x},\mathbf{z_1}),\mathbf{z_2}))))]. \end{aligned} \tag{17}$$

The content loss difference between this model and the based model is defined as

$$\mathcal{L}_D = \mathcal{L}_{L1}(G) - \mathcal{L}_{L1-G_{2D}}(G_{2D}). \tag{18}$$

As expected, the value of $\mathcal{L}_{L1-G_{2D}}(G_{2D})$ is lower than $\mathcal{L}_{L1}(G)$ as can be seen in Figure 11. However, different than Scenario-2, $\mathcal{L}_{L1-G_{2D}}(G_{2D})$ is smaller than $\mathcal{L}_{L1}(G)$ at every point but not at some interval (please refer to Figure 6 for comparison purpose). Since this model has more parameters than the based model in Scenario-1, training of this model (or reducing the content loss until the same value) should require more training iterations. However as can be seen in Figure, the loss value of this model is lower than the based model at every point. The reason of this is that, G_{1D} is receiving feedback from more resources via G_{2D} compared to G (in Scenario-1) and this helps faster training compared to based model.

For the sake of completeness, we also compare Scenario-4 with Scenario-2 in terms of the content loss in Figure 11a since they are closely related with each other. In order to monitor the difference, we defined $\mathcal{L}_{sc2-sc4}$ as

$$\mathcal{L}_{sc2-sc4} = \mathcal{L}_{L1-G_2}(G_2) - \mathcal{L}_{L1-G_{2D}}(G_{2D}). \tag{19}$$

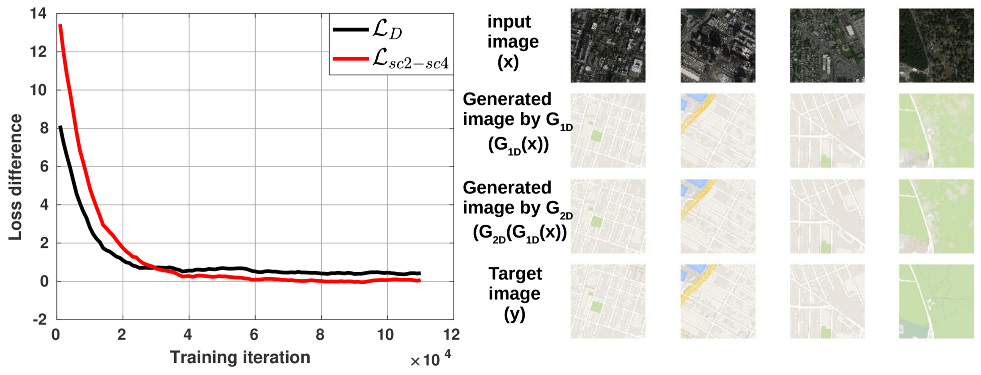


Figure 11. (a) Content loss differences. (b) Translated images by the model used in Scenario-4.

As can be seen from the figure, $\mathcal{L}_{L1-G_{2D}}(G_{2D})$ is smaller than the $\mathcal{L}_{L1-G_2}(G_2)$ at the initial training iterations because G_{1D} is receiving more feedback via G_{2D} compared to G_1 and therefore G_{1D} is trained faster than G_1. However at the final iterations, their values are almost the same and the difference is very close to 0 which indicates that using individual discriminator does not help a lot if enough training iterations are provided.

After this result, the final comparison of different scenarios' content losses at the final training iteration will be

$$\begin{gathered} \mathcal{L}_{L1-G_{2D}}(G_{2D}) \approx \mathcal{L}_{L1-G_2}(G_2) < \mathcal{L}_{L1}(G) < \mathcal{L}_{L1-G_B}(G_B) \\ scenario-4 \approx scenario-2 < scenario-1(base) < scenario-3, \end{gathered} \tag{20}$$

which indicates that although there is not a significant difference between using single or multiple discriminators, in terms of the content loss for the image-to-image translation settings, shrinking the conditional space helps improving the results. On the other hand, with a naive approach, by simply increasing the capacity of a single generator could be even worse. To quantitatively evaluate different scenarios more objectively, we computed per-pixel accuracy of the generated images as this method is commonly preferred in GAN community. The results are reported in Table 1. As can be seen from the results the difference between Scenario-2 and Scenario-4 is very small and both of them are better than Scenario-1, while Scenario-3 is the worst among them. Although the results do not break the record of the state-of-the-art results, it is sufficient to demonstrate the effectiveness of the proposed method.

Table 1. Per-pixel accuracy values of different scenarios on different datasets based on pix2pix.

Per-pixel acc	Scenario-1 pix2pix	Scenario-2	Scenario-3	Scenario-4
Aerial $\rightarrow$map	0.65	0.68	0.61	0.69
Facades $\rightarrow$labels	0.51	0.53	0.47	0.54

In order to validate our approach, we repeated same experiments presented at Scenario-1 through -4 for another dataset, Facades dataset [10] which consists of facades and corresponding segmentation of 506 buildings. The simulation results depicted in Figure 12 are consistent with the previous order of the content losses stated in Equation 20. Whereas it is quite subjective and there are only minor differences between the generated outputs in the Figure 12, visual inspection of them supports that Scenario-2 (sequential generators with one discriminator) and Scenario-4 (sequential generators with two discriminators) are roughly the same and better than the based line scenario-1(pix2pix). On the other hand, Scenario-3 (single but larger generator) is even worse than Scenario-1.

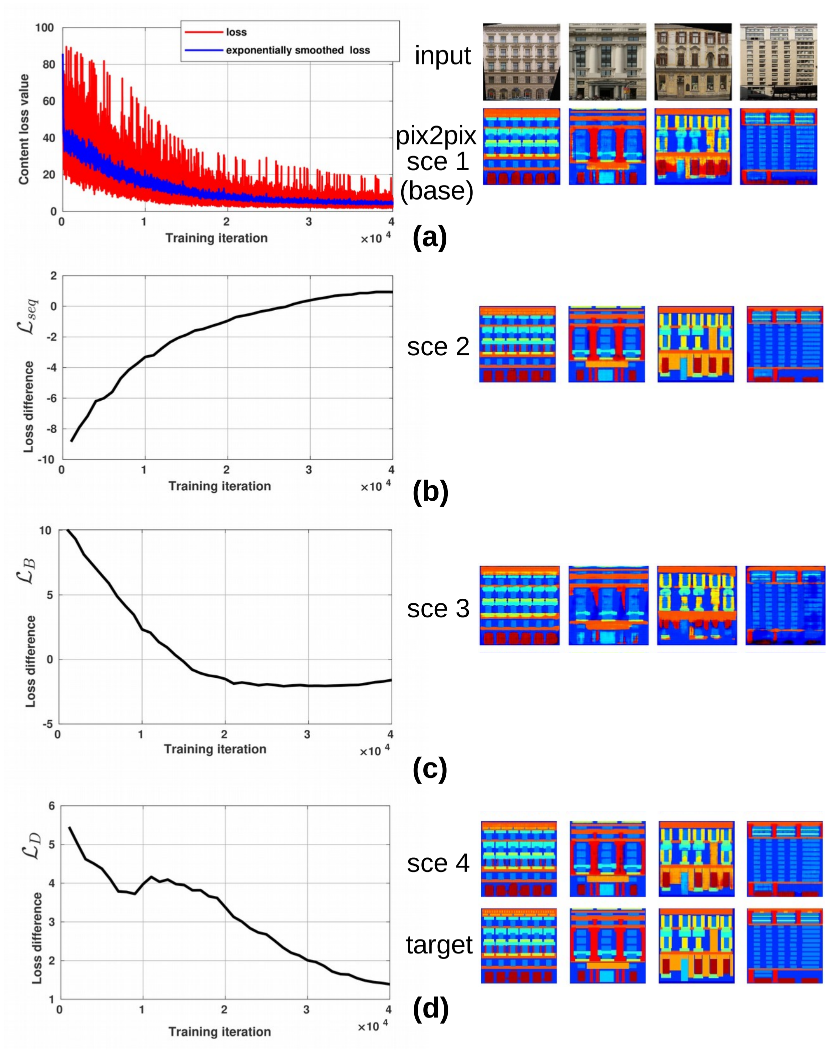


Figure 12. Content loss differences and the generated images for the same inputs for different scenarios.(a) Scenario-1, (b) Scenario-2, (c) Scenario-3, (d) Scenario-4.

For a further comparison, we repeat the same experiments for another image-to-image translation GAN model, CycleGAN [20]. In contrast to pix2pix model, CycleGAN is an unsupervised GAN model in the sense that, the input and output images should not be paired and can be chosen randomly from two different domains. For this reason, per-pixel acc. value of CycleGAN is lower than the value of pix2pix, as expected which can be seen in Table 2. That is being said, using sequential generators in the CycleGAN improves the results (Scenario-2) on both datasets, Maps and Facades. Per-pixel accuracy values jump to 0.61 and 0.49 from 0.55 and 0.42 for Maps and Facades datasets, respectively. In parallel to pix2pix, using a bigger generator instead of using sequential generators does not work well in CycleGAN also(Scenario-3). Under this scenario, per-pixel accuracy drops approximately 0.03 for both datasets. On the other hand, using two different discriminators for two generators (Scenario-4) can only slightly improve the results.

Table 2. Per-pixel accuracy values of different scenarios on different datasets based on CycleGAN.

Per-pixel acc	Scenario-1 CycleGAN	Scenario-2	Scenario-3	Scenario-4
Aerial →map	0.55	0.61	0.52	0.61
Facades →labels	0.42	0.49	0.39	0.50

4. Conclusion

Despite the fact that each GAN model has its own unique architectural design, some of the proposed models have a common approach which can be summarized as factorizing the difficult image generation process into smaller, more manageable subproblems. Although, this approach has demonstrated usefulness at the generated image quality, it is not yet quantitatively showed in an objective manner.

By utilizing the content loss we quantitatively demonstrate the effectiveness of factorizing the image generating process into multiple states. For this purpose, we take pix2pix GAN model, one of the image-to-image translation models, as based line and defined three different scenarios for different models and compare them with the based line model. In this first scenario, we evaluate the effect of shrinking the conditional space by connecting generators sequentially with one discriminator. In the second case, instead of using sequential generators we increase the capacity of single generator. In the final scenario, we design a GAN model with sequential generators and individual discriminators for each generator. Our simulation results indicate that in order to generate better images, simply increasing the capacity of a single generator might not help a lot (as depicted in Scenario-3) to reduce the content loss. On the other hand, shrinking the conditional space by using sequential generators significantly reduces the content loss as demonstrated at Scenario-2. In sequential generators, using multiple discriminators instead of one discriminator works better at the initial training iterations, however, if enough training iteration is provided, the content loss difference between these two models will be negligible as explained at Scenario-4.

References

[1] Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition; Salt Lake City, UT, USA; 2018. pp. 7132-7141.

[2] Tan M, Pang R, Le QV. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition; Seattle, WA, USA; 2020. pp. 10781-10790.

[3] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998 ;86 (11):2278-324.

[4] Dhariwal P, Jun H, Payne C, Kim JW, Radford A et al. Jukebox: A generative model for music. arXiv preprint arXiv:2005.00341.

[5] Xue A. End-to-end Chinese landscape painting creation using generative adversarial networks. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision; Virtual; 2021. pp. 3863-3871.

[6] Kingma DP, Welling M. Auto-Encoding Variational Bayes. Stat. 2014 ;1050: p.1.

[7] Kalchbrenner N, Espeholt L, Vinyals O, Graves A. Conditional image generation with PixelCNN decoders. Advances in Neural Information Processing Systems; Barcelona, Barcelona SPAIN; 2016. pp. 4797-4805.

[8] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D et al. Generative adversarial nets. Advances in neural information processing systems; Montréal, Montréal CANADA; 2014. 27.

[9] Kasgari ATZ, Saad W, Mozaffari M, Poor HV. Experienced deep reinforcement learning with generative adversarial networks (GANs) for model-free ultra reliable low latency communication. IEEE Transactions on Communications 2020;69 (2):884-899.

[10] Isola P, Zhu JY, Zhou T, Efros AA. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition; Honolulu, HI, USA; 2017. pp. 1125-1134.

[11] Nie W, Patel AB. Towards a better understanding and regularization of GAN training dynamics. In Uncertainty in Artificial Intelligence 2020;6 : 281-291.

[12] Ledig C, Theis L, Huszár F, Caballero J, Cunningham A et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE conference on computer vision and pattern recognition; Honolulu, HI, USA; 2017, pp. 4681-4690.

[13] Gui J, Sun Z, Wen Y, Tao D, Ye J . A review on generative adversarial networks: Algorithms, theory, and applications. IEEE Transactions on Knowledge and Data Engineering.

[14] Denton EL, Chintala S, Fergus R. Deep generative image models using a Laplacian pyramid of adversarial networks. Advances in Neural Information Processing Systems;Montreal, Quebec, Canada; 2015;28

[15] Wang X, Gupta A. Generative image modeling using style and structure adversarial networks. In European conference on computer vision; Amsterdam, The Netherlands; 2016: 318-335.

[16] Huang X, Li Y, Poursaeed O, Hopcroft JE, Belongie SJ. Stacked generative adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition; Honolulu, HI, USA; 2017.

[17] Zhang H, Xu T, Li H, Zhang S, Wang X et al . Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In Proceedings of the IEEE international conference on computer vision; Venice, Italy; 2017. pp. 5907-5915.

[18] Grnarova P, Levy KY, Lucchi A, Perraudin N, Goodfellow I et al. A domain agnostic measure for monitoring and evaluating gans. Advances in Neural Information Processing Systems; Vancouver, CANADA; 2019;32:12092-102.

[19] Fu SW, Liao CF, Tsao Y, Lin SD. Metricgan: Generative adversarial networks based black-box metric scores optimization for speech enhancement. In International Conference on Machine Learning; Long Beach, CA, USA; 2019,pp. 2031-2041.

[20] Zhu JY, Park T, Isola P, Efros AA. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision;Venice, Italy; 2017. pp. 2223-2232

[21] Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A et al. Improved techniques for training gans. Advances in neural information processing systems;Barcelona, Barcelona SPAIN; 2016.29:2234-42.

[22] Goodfellow Ian .Nips 2016 tutorial: Generative adversarial networks.arXiv preprintarXiv:1701.00160, 2016.

[23] Borji A. Pros and cons of GAN evaluation measures: New developments. Computer Vision and Image Understanding. 2022 ;215 (1):103329.

[24] 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. Advances in neural information processing systems; Long Beach, CA, USA; 2017.30.

[25] Karacan L, Akata Z, Erdem A, Erdem E. Learning to Generate Images of Outdoor Scenes from Attributes and Semantic Layouts. arXiv e-prints. 2016 Dec:arXiv-1612.

[26] Yi Z, Zhang H, Tan P, Gong M. Dualgan: Unsupervised dual learning for image-to-image translation. In Proceedings of the IEEE international conference on computer vision;Venice, Italy; 2017 pp.2849-2857

[27] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention;Munich, Germany; 2015. pp. 234-241

[28] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning;Lille, France; 2015, pp. 448-456

[29] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research. 2014; 15 (1):1929-58.