

# Deep Generative Learning for the Generation and Analysis of Architectural Plans with Small Datasets

David Newton<sup>1</sup>

<sup>1</sup>University of Nebraska-Lincoln College of Architecture

<sup>1</sup>david.newton@unl.edu

*The field of generative architectural design has explored a wide range of approaches in the automation of design production, but these approaches have demonstrated limited artificial intelligence. Generative Adversarial Networks (GANs) are a leading deep generative model that use deep neural networks (DNNs) to learn from a set of training examples in order to create new design instances with a degree of flexibility and fidelity that outperform competing generative approaches. Their application to generative tasks in architecture, however, has been limited. This research contributes new knowledge on the use of GANs for architectural plan generation and analysis in relation to the work of specific architects. Specifically, GANs are trained to synthesize architectural plans from the work of the architect Le Corbusier and are used to provide analytic insight. Experiments demonstrate the efficacy of different augmentation techniques that architects can use when working with small datasets.*

**Keywords:** generative design, deep learning, artificial intelligence, generative adversarial networks

## INTRODUCTION

Generative architectural design is a field of study that explores the automation of building design in whole, or in part. Although a generative design process need not be done with a computer, the majority of current research focusses on the development of algorithms that can be run on a computer to create designs. Researchers in the field have explored a variety of algorithmic approaches that run with varying levels of autonomy in the creation of designs. In more autonomous roles, developed algorithms take on the role of primary author with little user input. In more collaborative roles, they work interactively with architects to generate designs. Previous work in these areas has been limited in its ability to demon-

strate significant artificial intelligence in which generative algorithms are able to learn from and interpret experience to adaptively meet goals (Russell and Norvig 2010). Recent advancements in the field of Machine Learning (ML), however, open the door to significantly improve the intelligence of generative algorithms.

Deep Neural Networks (DNNs) have emerged as a leading approach in ML for both discriminative and generative learning tasks. Deep discriminative models have demonstrated the ability to outperform human experts on classification and recognition tasks, while deep generative models have outperformed competing approaches in the synthesis of images. Generative Adversarial Networks (GANs) (Goodfellow

et al. 2014) are a leading deep generative model that have demonstrated impressive results on 2D and 3D design tasks. Their exploration in the field of architecture, however, has been limited to the generation of generic architectural plans and facades (As, Pal, and Basu 2018, Huang and Zheng 2018, Isola et al. 2017).

This research addresses this gap in previous work and contributes knowledge on the application of GANs in the generation and analysis of architectural plans from specific stylistic movements in architecture. Specifically, a GAN is trained on a selection of residential plans by the architect Le Corbusier. The GAN is then used to synthesize new 2D plan images in the style of Corbusian planning. A series of experiments test different techniques architects might use when working with small datasets. They show that augmentation strategies can be effectively used with small training sets to improve image quality. Lastly, the research demonstrates how GANs can be used by architects to provide analytic insight into the deep organizational structure of specific architectural oeuvres.

## BACKGROUND

Previous research on generative architectural design can be divided into five major categories: optimization and search; physically-based; generative grammars; general probabilistic; and deep generative models. Optimization and search algorithms have been used to synthesize a variety of designs from architectural plans (Caldas and Santos 2012), to building envelopes (Gagne and Andersen 2010, Turrin, von Buelow, and Stouffs 2011). Physically-based algorithmic approaches have used physics simulation (Klemmt and Bollinger 2017) and growth models inspired by nature to create architectural forms (Dincer, Cagdas, and Tong 2014). Approaches based on generative grammars (Stiny 1980) use the iterative application of a set of rewriting rules to synthesize designs, and have been used to generate architectural plans (Stiny and Mitchell 1978), master plans (Halatsch, Kunze, and Schmitt 2008), and facades (Müller et al. 2006). General probabilistic approaches have

explored the application of Bayesian networks (Merrell, Schkufza, and Koltun 2010) and Markov chains (Swahn 2018) to learn from example images in the synthesis of architectural and urban plans respectively. All these approaches surveyed demonstrate little to no artificial intelligence by the definition provided by Russell and Norvig (2010), who are leading researchers in the field of AI. Specifically, they demonstrate little ability to learn from and interpret experience adaptively towards the synthesis of architectural designs.

The last category of generative architectural design research explores the use of deep generative models. In contrast to other generative approaches, deep generative models demonstrate an impressive ability to learn from a set of examples to synthesize designs. Generative Adversarial Networks (GANs) are a leading deep generative model that use two competing DNNs in order to learn the probability distribution that underlies the organizational structure of a set of training examples (Goodfellow et al. 2014). Once this probability distribution is learned, the GAN can synthesize new example instances. For example, GANs have been used to generate new designs of bedrooms based on a collection of photographs of bedroom interiors (Radford, Metz, and Chintala 2015). They have also learned to synthesize 3D designs of furniture by learning from a training set of example 3D models of furniture (Wu et al. 2016). Their use in the field of architecture, however, has been limited to the generation of generic architectural plans (As, Pal, and Basu 2018, Huang and Zheng 2018) and a variety of facades from different architectural styles (Isola et al. 2017). In contrast to the use of GANs with generic, or highly diverse datasets, this research will contribute to this previous work by applying GANs to the generation and analysis of a specific architect's work. The residential plans of Le Corbusier.

## METHODOLOGY

A key challenge in working with GANs is procuring enough training examples. Typical training set sizes

Table 1  
Shows the training parameters for the four experiments conducted.

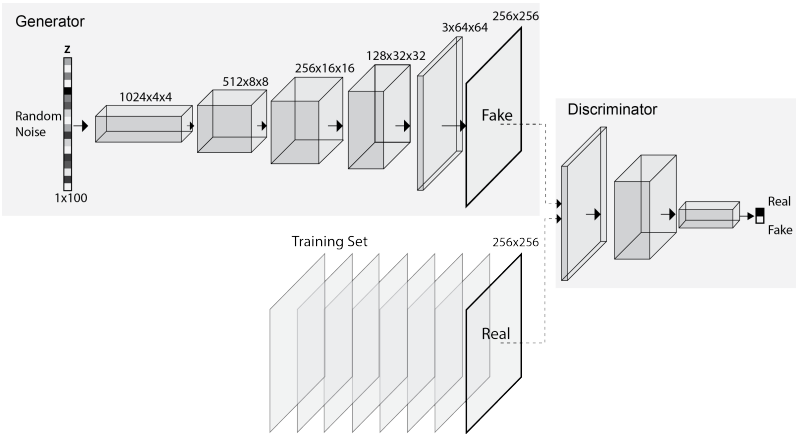
Generative Task	GAN Used	Training Set Size	Optimizer	Learning Rate: Generator / Discriminator	Epochs	Training Time	Batch Size	Augmentation	Input/Output Resolution
2D Plans – Le Corbusier	WGAN	45	ADAM	0.002 / 0.00005	1000	0.5 hours	32	None	256 x 256 / 256 x 256
2D Plans – Le Corbusier	WGAN	135	ADAM	0.002 / 0.00005	1000	1.5 hours	32	Noise	256 x 256 / 256 x 256
2D Plans – Le Corbusier	WGAN	180	ADAM	0.002 / 0.00005	1000	2 hours	32	Rotation;	256 x 256 / 256 x 256
2D Plans – Le Corbusier	WGAN	540	ADAM	0.002 / 0.00005	1000	8 hours	32	Noise; Rotation;	256 x 256 / 256 x 256

for peak performance in image synthesis tasks can contain 10,000 to 50,000 images (Im et al. 2018). Architectural datasets, in contrast, can be much smaller. How do GANs perform on small datasets? What techniques can be used to augment small datasets and what effects does this have? This research attempts to answer these questions by working with a small dataset comprised of 45 plan images from a variety of Le Corbusier’s single-family residential projects. Four different experiments are run on this dataset to test the effects of different augmentation strategies as shown in Table 1: test one uses no augmentation; test two uses augmentation with noise (i.e., producing additional training examples by adding Gaussian, Poisson, and random noise to an original image); test three uses augmentation with rotation of the original images in 90 degree increments: test four uses

noise and rotation. The generated plans produced from these experiments are then visually assessed and qualitatively compared.

There are a number of GAN architectures available for 2D image synthesis for architects to work with. For this research, the Wasserstein Generative Adversarial Network (WGAN) (Arjovsky, Chintala, and Bottou 2017) was selected, because of its relative stability during training, as compared to other GAN variants. Figure 1 shows a diagram of the WGAN’s architecture - which is typical of most GANs. In that architecture, a generator DNN and a discriminator DNN compete with one another. The job of the discriminator network is to tell the difference between fake images created by the generator network and real images from the training dataset. The generator network, meanwhile, has the job of trying to synthesize

Figure 1  
The diagram shows the generator and discriminator networks that comprise the WGAN.



images that will fool the discriminator network. This competition allows the generator network to build a model of the probability distribution that underlies the training examples, and then to sample from that model to create new instances that resemble the training set.

This WGAN architecture was implemented using the Python programming language with TensorFlow. The GAN was trained for 1000 epochs on a PC with one NVIDIA GTX 1080 GPU. An epoch refers to the number of training cycles in which the dataset is passed through the GAN, and the number of epochs was chosen after initial experiments showed that improvement stopped at around this time for most models. The detailed training parameters can be seen in Table 1.

### RESULTS AND ANALYSIS

The training losses of the WGAN's generator and discriminator networks for all four experiments can be seen in Figure 2. Part a of the figure shows the results of training with no augmentation applied to the dataset. The loss graph indicates that the discriminator network quickly learns how to outperform the generator network, and that the generator's performance fails to improve for most of the training except at the very end. In part b of the figure, the training losses are shown when the dataset is augmented using the addition of noise. This augmentation technique seems to improve the performance of the generator network and its loss continues to get smaller throughout the training. Parts c and d of the figure show the results for augmentations using rotation and rotation with noise respectively. Both of these options show a longer trajectory of decreasing generator performance. These experiments show that the noise augmentation is most successful of the four approaches in reducing the loss of the generator. Further, it shows that the size of the dataset did not seem to be the determining factor in the training. The reasons for this are unclear and require further study.

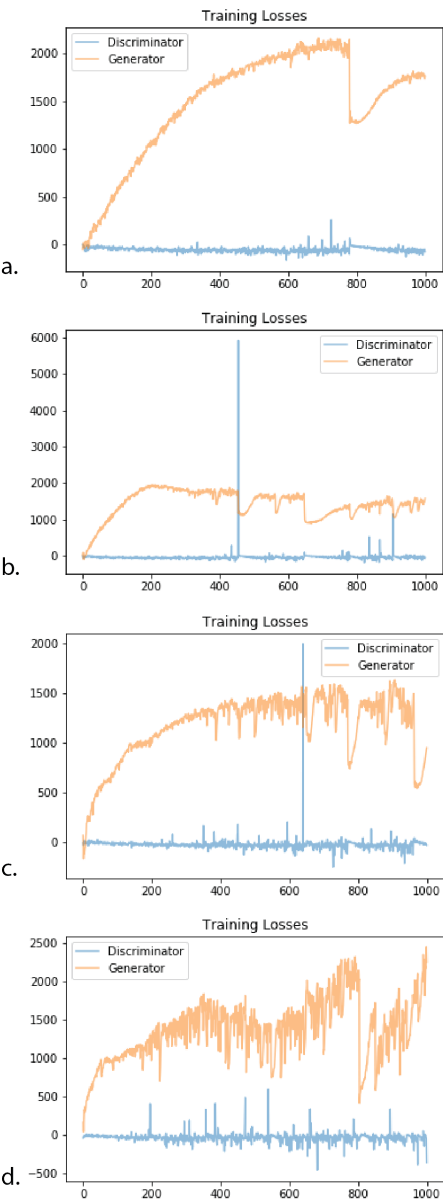


Figure 2  
Shows the training losses for the generator and discriminator networks for the following experiments: a) No augmentation is applied; b) Augmentation in the form of noise is applied; c) Augmentation in the form of rotation is applied; d) Augmentation in the form of noise and rotation is applied.

Figure 3

Part a-c show samples of GAN generated plans using the original 45 image Le Corbusier house dataset with no augmentation. Part d-f show samples generated with noise augmentation. Part g-i show samples generated using rotation. Part j-l show samples generated using noise and rotation as augmentation strategies.

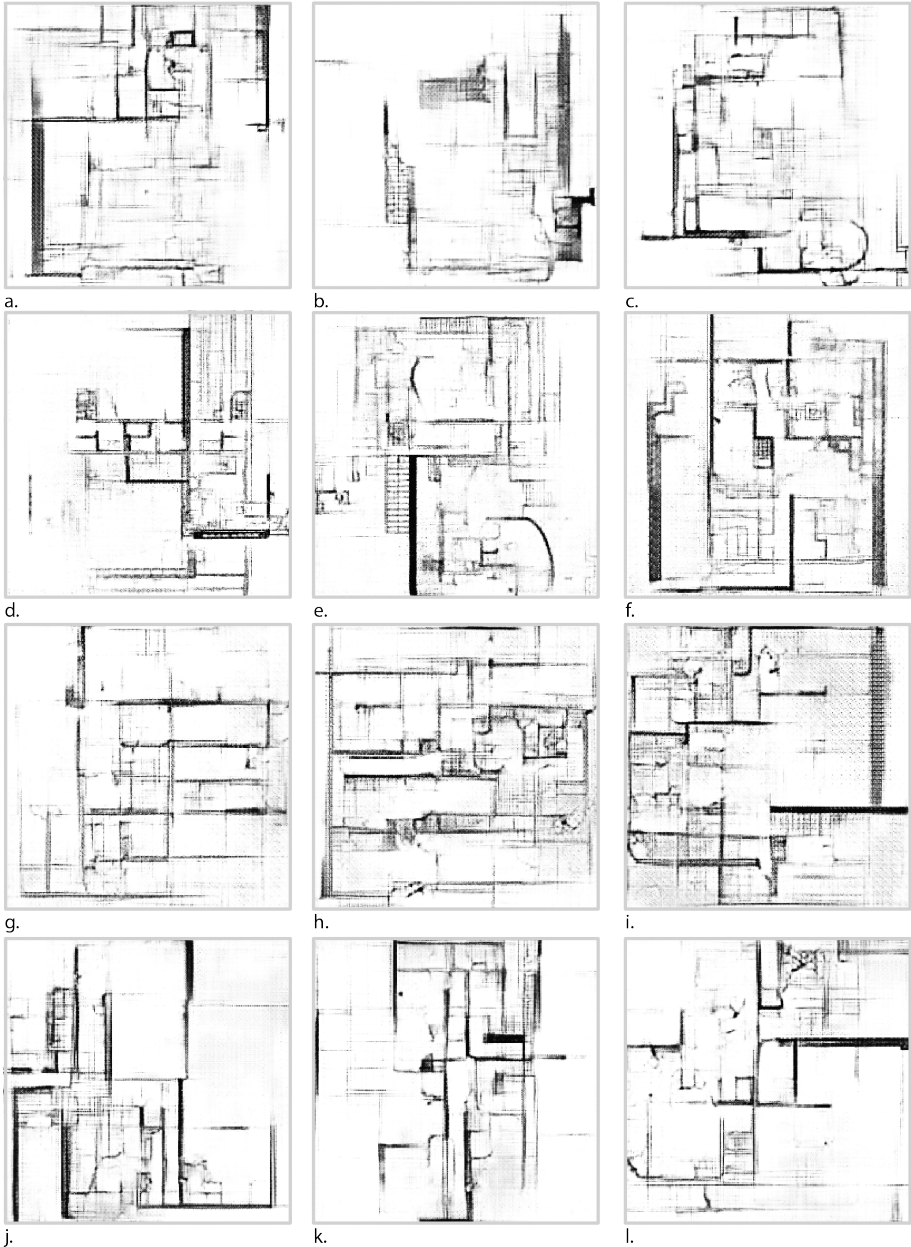


Figure 3 shows GAN generated samples from the three experiments outlined in Table 1. In parts a-c of the figure, samples generated with the original dataset appear fuzzy. This low fidelity is the result of the poor training performance of the generator as show in Figure 2 part a. In parts d-f of Figure 3, samples generated using noise augmentation can be seen. This simple augmentation improves the visual fidelity of the generated images to the point where stairs can begin to be seen. In parts g-i of the figure, samples generated using rotation can be seen. These samples seem to be visually equivalent in quality to the samples generated with no augmentation. Parts j-l of the figure, then show samples generated using both noise and rotation augmentations. The visual quality of these samples, despite the dataset being 12 times larger than the smallest dataset, seems equivalent to the samples produced with no augmentation. The visual quality of the highest performing augmentation approach still creates images with low visual quality, in addition to producing designs which may not practically work. This may be due to the small training set size and the chosen GAN architecture. Huang and Zheng (2018), for example, use the information GAN (Isola et al. 2017) architecture to generate high resolution architectural plans.

In addition to generating new images, a trained GAN can also provide analytic insight into the training set. This is due to the fact that, through the training, the GAN learns to recognize the low and high-level image features that define the essential qualities of the training set. These learned features can be viewed by looking at the activation patterns of the neural network layers that comprise the generator and discriminator networks of the GAN. Figure 4 shows activation patterns from selected neural network layers within the discriminator network of the GAN trained with noise augmentations. Layer one of the figure shows low-level features like horizontal and vertical edges being learned - represented by the brighter pixels in the activation map. Layer two of the figure, shows higher level features such as walls, windows, and cabinetry being learned. In layer

three, more complex features such as the larger organizing grid of the house is being learned. Figure 5 then shows samples of the neural network layers in the generator network of the GAN - which work in the reverse order of the discriminator. Both networks can provide insight into the deep organizational structure of Le Corbusier's house plans, but the activation patterns learned by the DNN still require careful study and interpretation. Further, because there are often hundreds of neural layers, it can be difficult to know what each has learned.

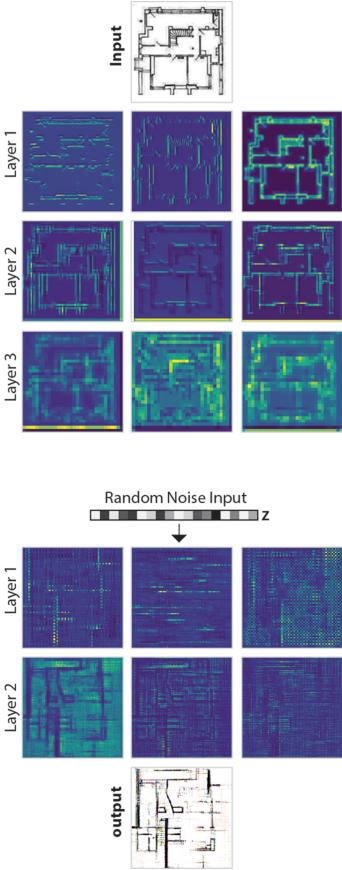


Figure 4  
This figure shows activation patterns from selected neural network layers within the discriminator network of the GAN trained with noise augmentations.

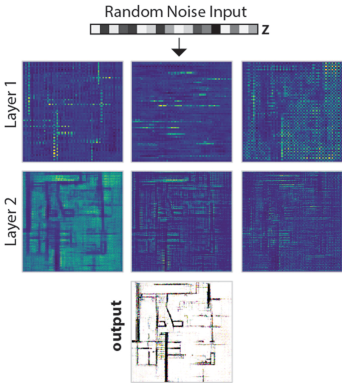


Figure 5  
This figure shows the activation patterns of select neural network layers from the generator network of the GAN trained with noise augmentations.

## CONCLUSIONS

This research contributed new knowledge on the application of GANs in the synthesis and analysis of architectural plans in the style of a particular architect. Specifically, experiments were conducted that demonstrated the use of GANs in the generation of single-family house plans based on the work of Le Corbusier. These experiments demonstrated the efficacy of different data augmentation techniques when working with small datasets. The experimental results indicated that noise augmentation outperformed competing approaches.

This research also pointed the way towards a number of areas for future investigation. One area for such development deals with the use of GANs for analytic insight. There are a number of ways to visualize the neuronal layers and the filters that are part of a GAN. But which visualization strategy, or strategies, might be most useful for different types of architectural analysis? Further, how do different GAN architectures compare in terms of providing such insight? Are some better at providing interpretable representations than others? What are the evaluation criteria that researchers might use to compare models in this way? The answer to these questions and others, in relation to how GANs might be integrated into the architectural design process, loom large as the discipline seeks to stay relevant in this AI-driven future.

## REFERENCES

- Arjovsky, M, Chintala, S and Bottou, L 2017, 'Wasserstein gan', *arXiv preprint arXiv:1701.07875*, 1, pp. 5-7
- As, I, Pal, S and Basu, P 2018, 'Artificial intelligence in architecture: Generating conceptual design via deep learning', *International Journal of Architectural Computing*, 16(4), pp. 306-327
- Caldas, LG and Santos, L 2012, 'Generation of Energy-Efficient Patio Houses With GENEARCH', *Combining an evolutionary generative design system with a shape grammar*, eCAADe, 2, pp. 36-45
- Dincer, AE, Cagdas, G and Tong, H 2014, *A digital tool for customized mass housing design*, Newcastle upon Tyne
- Gagne, JM and Andersen, M 2010, 'Multi-objective facade optimization for daylighting design using a genetic algorithm', *Proceedings of SimBuild*, 4(1), pp. 110-117
- Goodfellow, I, Pouget-Abadie, J, Mirza, M, Xu, B, Warde-Farley, D, Ozair, S, Courville, A and Bengio, Y 2014 'Generative adversarial nets', *Advances in neural information processing systems*, pp. 2672-2680
- Halatsch, J, Kunze, A and Schmitt, G 2008, 'Using shape grammars for master planning', in John S. Gero, JG and Ashok k. Goel, AG (eds) 2008, *Design Computing and Cognition*, Springer, pp. 655-673
- Im, DJ, Ma, H, Taylor, G and Branson, K 2018, 'Quantitatively evaluating GANs with divergences proposed for training', *arXiv preprint arXiv:1803.01045*, 1, pp. 2-4
- Isola, P, Zhu, JY, Zhou, T and Efros, AA 2017 'Image-to-image translation with conditional adversarial networks', *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125-1134
- Klemmt, C and Bollinger, K 2017, 'Angiogenesis as a model for the generation of load-bearing networks', *International Journal of Architectural Computing*, 15(1), pp. 18-37
- Merrell, P, Schkufza, E and Koltun, V 2010 'Computer-generated residential building layouts', *ACM Transactions on Graphics (TOG)*, p. 181
- Meuller, P, Wonka, P, Haegler, S, Ulmer, A and Van Gool, L 2006, 'Procedural modeling of buildings', *Acm Transactions On Graphics*, 25(3), pp. 614-623
- Radford, A, Metz, L and Chintala, S 2015, 'Unsupervised representation learning with deep convolutional generative adversarial networks', *arXiv preprint arXiv:1511.06434*, 1, pp. 1-5
- Russell, SJ and Norvig, P 2016, *Artificial intelligence: a modern approach*, Malaysia; Pearson Education Limited,
- Stiny, G 1980, 'Introduction to shape and shape grammars', *Environment and planning B: planning and design*, 7(3), pp. 343-351
- Stiny, G and Mitchell, WJ 1978, 'The palladian grammar', *Environment and planning B: Planning and design*, 5(1), pp. 5-18
- Swahn, E 2018 'Markovian Drift - Iterative substitutional synthesis of 2D and 3D design data using Markov models of source data', *Computing for a better tomorrow - Proceedings of the 36th eCAADe Conference*, Lodz University of Technology, Lodz, Poland, 19-21
- Turrin, M, Von Buelow, P and Stouffs, R 2011, 'Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms', *Advanced Engineering Informatics*, 25(4), pp. 656-675

Wu, J, Zhang, C, Xue, T, Freeman, B and Tenenbaum, J  
2016 'Learning a probabilistic latent space of object  
shapes via 3d generative-adversarial modeling', *Ad-  
vances in neural information processing systems*, pp.  
82-90