GAN-based method for generative design of visual comfort in underground space

Yingbin GUI¹², Biao ZHOU¹²*, Xiongyao XIE¹², Wensheng LI³, Xifang ZHOU⁴

1 Key Laboratory of Geotechnical and Underground Engineering, Tongji University
2 Department of Geotechnical Engineering, Tongji University
3 China Railway Siyuan Survey and Design Group
4 Shanghai City Road Investment (Group) Co. Ltd

Corresponding author’s e-mail: zhoubiao@tongji.edu.cn
Corresponding author’s ORCID: https://orcid.org/0000-0003-2083-2456

Abstract. Existing methods for the generative design of architecture mainly focus on layout planning, and the genetic algorithm is popular in related research. However, the genetic algorithm is not available for non-quantitative problems (e.g., visual comfort) and inefficient for complex problems. In response, recent studies used the generative adversarial network (GAN) as the generative model to realize efficient generation. However, existing GAN-based methods for generative design are not effective enough because the output of their models is a two-dimensional image, which must be reconstructed to a three-dimensional (3D) model in practical engineering applications. In this study, a visual comfort generative network (VCGN) based on parametric modeling and a semi-supervised generative adversarial network (SGAN) is proposed for a more effective generation. The VCGN contains three parts, namely the parametric and teacher models and the SGAN. The parametric model is designed to generate a 3D underground space model by controlling the key parameters. The teacher model rates the visual comfort level of the generated model and trains the SGAN and could be a neural network or another criterion. The SGAN is designed to learn parameter distribution and generate parameters. Different from the SGAN mentioned by Odena, the SGAN used herein comprises four parts: two generators, a discriminator, and a classifier. The first generator is designed to generate parameters, while the second is used to simulate the parametric modeling process. The discriminator is supposed to distinguish simulated results from real results. The classifier is designed to classify the visual comfort levels. Moreover, a spatial color-generation task in underground space is applied as a case study to validate the effectiveness of the VCGN. A comparison of the random sampling method and the existing GAN-based method shows that the VCGN could generate more high-comfort underground space models than other methods for the same amount of time. In addition, a comparison of the underground space models under different comfort levels generated by the VCGN illustrates that the VCGN has the potential to generate an underground space model with a specified comfort level.

1. Introduction
The current acoustic, thermal, and wet environments in the underground space are close to those aboveground; however, the visual comfort of the underground space is insufficient. Previous studies [1,2] have considered color, scale, brightness, and natural elements to be the key factors of the visual comfort in the underground space. However, these visual comfort factors are subjective and difficult to quantify. Therefore, evaluating the visual comfort of space and feedback design have remained challenging.

Generative design is a powerful design method considered to be an algorithm- or rule-based method of generating multiple and complex solutions [3]. A generative design algorithm consists of evaluation and generative algorithms. The existing methods for the generative design of architecture mainly focus on layout planning. Accordingly, the genetic algorithm (GA) is very popular in related research. The GA was first applied in architecture design by L. Caldas et al. [4]. Following their research, the GA was applied in green design [5], energy consumption [6], and other building design problems [7]. Despite its advantages, however, the GA is not suitable for subject problems and inefficient in complex problems.

To address the abovementioned issue, studies have applied the generative adversarial network (GAN) as the generative algorithm in generative design to achieve a more efficient generation. The GAN is composed of a generator and a discriminator. The generator generates a fake sample, while the discriminator distinguishes true samples from false ones. The GAN was first applied in a planning problem in architecture [8,9], and its output was an image with noise, which was hard to apply in engineering. Correspondingly, a three-dimensional (3D) IWGAN has been employed to generate a 3D architecture model [10]. Many 3D models for training the 3D IWGAN model can be obtained online. However, the training process is time consuming, and the 3D dataset is hard to establish. Moreover, the GAN has always encountered the common problem of a noisy output, and the 3D GAN is not an exception.

The existing GAN-based methods aim to solve problems in layout planning, not visual comfort design. The generative design of visual comfort presents less related studies. Furthermore, GAN-based methods have not been applied to solve the generative design of visual comfort in the underground space. Hence, how to design the visual comfort of the underground space has remained a challenge, and how to use GAN to solve this problem needs further study.

Therefore, we explore herein a new generative design framework and propose a visual comfort generative network (VCGN) based on parametric modeling and a semi-supervised generative adversarial network (SGAN). We validate the effectiveness of VCGN by applying a spatial color-generation task in the underground space as a case study. By comparing the random sampling method and the existing GAN-based method, we found that the VCGN could generate more high-comfort underground space models than other methods for the same amount of time. A comparison of the underground space models under different comfort levels generated by the VCGN also showed that the VCGN has the potential to generate an underground space model with a specified comfort level.

2. Methodology

The GAN covers an extensively large research field, and many related studies have been presented in recent years. In this study, the SGAN was adopted and refined to fit our framework. We will introduce SGAN and VCGN below, as well as the detailed network structure and the functions of each part.

2.1. SGAN

The GAN was originally proposed by Goodfellow [11]. It aims to find the balance point between a generator and a discriminator. The initial GAN is a minimax game with the value function of the generator and the discriminator. As shown in Eq. (1), in the training process, the discriminator is trained to maximize the probability of distinguishing true samples from fake ones, while the generator is trained to cheat on the discriminator and minimize the \( \log(1 - D(G(z))) \).

\[
\min_{G} \max_{D} V(D, G) = \mathbb{E}_{x \sim p_{data}} \left[ \log D(x) \right] + \mathbb{E}_{z \sim \rho(z)} \left[ \log(1 - D(G(z))) \right] 
\]  

(1)
In the application, the network structure that can generate samples does not meet the demand. The applicability of the GAN will be greatly improved if it can generate a sample with a specified category. To solve this problem, a conditional generative network (CGAN) has been proposed [12]. The inputs of the generator and the discriminator are samples with labels; however, the output of latter is only a variable representing true or false. There is no classification loss function in the CGAN training. The CGAN performance is not very good in application.

The SGAN was proposed by A. Odena [13] and T. Salimans [14] and was supposed to classify the sample while distinguishing it. As shown in Figure 1, compared with the CGAN, the last layer of the discriminator is replaced with a SoftMax layer in the SGAN, and the input of SGAN’s generator is sampled without a label. The SGAN output is an \( N + 1 \) dimension vector that contains \( N \) classes and a variable representing false. The loss function for classifier training is as follows:

\[
L = -E_{x,y \sim p_{data}(x,y)}[\log p_{model}(y \mid x)] - E_{x \sim \mathcal{G}}[\log p_{model}(y = N + 1 \mid x)]
\]

2.2. VCGN
The VCGN is a network that combines parametric modeling and the SGAN. A previous research [7] has discussed the benefits of combining parametric modeling and genetic algorithms, but no GAN-based method has yet combined parametric modeling and the GAN. As a powerful generative model, the GAN can solve more problems (e.g., visual comfort) compared to the GA. In addition, its combination with parametric modeling makes it easier to be applied in engineering. Parametric modeling and the SGAN are integrated well in the VCGN. An evaluation model has also been added to the network to conduct semi-supervised learning.

Figure 2 shows that the VCGN contains three main parts: parametric modeling, teacher model, and SGAN. The parametric model is designed to generate a 3D underground space model by controlling the key parameters. The teacher model rates the visual comfort level of the generated models and trains the SGAN. The SGAN is designed to learn parameter distribution and generate parameters. In the training process, a batch of random noises is first generated and input into the SGAN generator. Subsequently, the generator generates the parameters, which are the input of parametric modeling and the simulator. Next, parametric modeling generates a 3D model and outputs a scene image. The teacher model then evaluates the comfort level of the scene image. Finally, the
scene image with a comfort level and the simulator output are used to train the discriminator. In the test process, a random noise is generated, and the generator generates parameters based on this noise. Parametric modeling then generates 3D models and images. Finally, the teacher model evaluates all the images and selects the models that meet the requirements.

Parametric modeling enables the VCGN to be embedded in the BIM (Building Information Modeling) platform. The main program in the first version of parametric modeling was developed in Revit. However, Revit is a commercial software, and the render function cannot be accessed through its API; thus, the parametric modeling program was developed in Blender, a powerful open-source modeling software. The parametric modeling program contains several functions, including initialization, component creation, adding material, adding texture, changing the parameters, and rendering the function. Figure 3 depicts the detailed process of the parametric modeling, which is also presented below:

- **Initialization**: The first step is to establish the spatial relation of different components and define the components that must be placed in different positions.
- **Component generation**: Generate the components after the location of different components is determined. The related components contain a column, a floor, a roof, a wall, and a surface layer.
- **Adding material and texture**: We only add material in this study.
- **Changing parameters**: This step can be skipped if the model is being built for the first time; otherwise, the program will change the related parameters based on the input. In theory, the program can change the parameters of any model part. We only changed the color of a different material herein.
- **Rendering**: Blender provides two types of rendering: Cycles and EEVEE (Extra Easy Virtual Environment Engine). Cycles is a very powerful render; hence, we employed it to make the scene image more real.
The teacher model labels the scene image and filters the 3D model. It does not participate in error backpropagation; therefore, machine learning algorithms or criteria can also serve as a teacher model. The teacher model requirements are high due to its role. When used as the criteria, we consider their accuracy to be 100%. The accuracy of a machine learning algorithm is supposed to be greater than 90% when employing a machine learning algorithm as the teacher model.

In this study, we employed a trained random forest algorithm as the teacher model. Figure 4 shows that the scene image is split directly into parts. We split the image into $15 \times 15$ parts here. The features then functioned to extract the features of each part. We studied the color configuration of the underground space; hence, we directly extracted the color model of each part. Finally, the random forest algorithm was used to predict the comfort level of the image. The random forest algorithm was used in the binary classification tasks to ensure the evaluation accuracy. The accuracy was approximately 94%.

Different from the SGAN mentioned by A. Odena [13] and T. Salimans [14], we added a simulator to the SGAN architecture. In the training process, the parametric modeling is not a neural network; thus, it cannot perform error backpropagation, and the generator cannot refresh its weights. From the perspective of the input and the output of the parametric modeling, it can be seen as $f(x)$, where $x$ represents the parametric modeling input. This means that the whole process is subject to the chain rule, but we cannot know the exact $f(x)$ formula. Therefore, we added a simulator to the SGAN architecture to determine the $f(x)$ formula and backpropagate the error.

From the perspective of the GAN structure, the simulator serves as a part of the generator. The generator and the simulator can be combined into a bigger generator. The two parts calculate the gradient as a whole in the backpropagation process. In addition, the simulator output should be the features extracted from the scene image. If the output is a scene image, each layer of the simulator must be replaced with a convolutional layer. Although we can add a convolutional layer to the simulator based on the deep convolutional generative adversarial networks (DCGAN) principle [15], the fully connected layer of the generator cannot be replaced, resulting in an unstable SGAN training. Figure 5(a) illustrates that in our attempt, if we replace the simulator and the discriminator with the DCGAN structure, the generator output always comprises bad comfort level parameters referred to as
a mode collapse. On the contrary, without replacement, the SGAN training process is very stable, and the generated image becomes highly comfortable.

In our view, the main reason for this phenomenon is the unbalanced generator and discriminator. In the architecture guidelines of the DCGAN, the fully connected layers are removed. The generator and the simulator were separated when we replaced the simulator and the discriminator with convolutional layers. On the contrary, the simulator and the generator were combined into a bigger discriminator, and the weight parameter of the discriminator was far more than that of the generators. The discriminator weight update was more effective than that of the generator. In this case, the generator and the discriminator did not reach the Nash equilibrium.

![Figure 5](image1.png)

**Figure 5.** Training process of different simulators and discriminators: (a) with the DCGAN architecture and (b) with our SGAN.

![Figure 6](image2.png)

**Figure 6.** Different comfort levels of the generated underground space.

### 3. Case study and comparison

To test the effectiveness of the VCGN, a spatial color-generation task was applied, and a simple 3D underground space model was built. We hope that the resolution of the rendered scene is as high as possible; thus, four GeForce RTX 2080Ti GPUs were used to complete the rendering process. Meanwhile, only one GeForce RTX 2080Ti GPU was used in the SGAN training process. We used the Intel Core i9-9940X CPU in our workstation.

Figure 6 shows that the VCGN can generate high- and low-comfort level underground space models, and the color scheme of the generated underground space is not common in daily life. This generation can help inspire designers match different colors, and better color schemes are sometimes
produced. Compared with the image generated by the GAN, the scene image generated by the VCGN has no noise point because the underlying principles of our image generation are different. In the traditional GAN, the image is fully generated by the neural network, and the process cannot be 100% accurate. In the VCGN generation, the image is generated by scene rendering, which is much more accurate than a neural network. The generated underground space model is also 3D and can be directly used in engineering.

We validated the effectiveness of the VCGN by comparing it with previous methods. The random sample method randomly generates parameters, and its generation process is unable to control. Figure 5(b) depicts that compared with the random sample method, the VCGN generation is developing in the direction of high comfort. Compared with the existing GAN-based methods, the VCGN generation is a 3D model that can be used in engineering. We also calculated the time for the two methods to generate to further show the difference in the efficiencies of the VCGN and the GAN-based methods. In the existing GAN-based methods, the image is generated at approximately 0.1 s. However, it takes about 10 min to modify the 3D model of the subterranean space based on the images, and the total time spent is 600.1 s. Meanwhile, in the VCGN generation, the total time for the model modification and the image rendering is approximately 12 s. As far as the overall modeling process is concerned, the VCGN efficiency is higher than that of the existing GAN-based methods.

4. Conclusion
We proposed herein the visual comfort generative network and describe its detailed implementation process. The VCGN contains parametric modeling, a teacher model, and the SGAN. The importance and the role of the teacher model and the parametric modeling were clearly explained by describing their processes. By comparing the improved SGAN with the DCGAN architecture, we found that the former is more stable and suitable for generating an underground space model. We also analyzed the possible causes of the abovementioned phenomenon, which can provide reference for a follow-up related research. By applying a spatial color-generation task, we proved that the VCGN can generate high- and low-comfort level underground space models. Comparing the existing GAN-based methods and the random sample method, we found that the VCGN efficiency is higher than that of the others. The VCGN generates a 3D model and is more practical. However, the VCGN generation also has some problems. In the case study, we found that the rendering process requires hardware support, and the entire process consumes computing resources. Moreover, although the VCGN can generate a highly comfortable underground space model, the diversity of the generated underground space is insufficient. To solve these problems, the optimization of the GAN model and the modeling process will be performed in a future research.

Acknowledgments
The work is supported by the National Key Special Project (2018YFC0808702), National Natural Science Foundation of China (52038008), and Shanghai Science and Technology Commission Innovation Action Plan (20dz1202406).

References
[1] Yildirim K, Akalin-Baskaya A and Hidayetoglu M L 2007 Effects of indoor colour on mood and cognitive performance Build. Environ. 42 3233–40
[2] Duff J, Kelly K and Cuttle C 2017 Spatial brightness, horizontal illuminance and mean room surface exitance in a lighting booth Light. Res. Technol. 49 5–15
[3] Caetano I, Santos L and Leitão A 2020 Computational design in architecture: Defining parametric, generative, and algorithmic design Front. Archit. Res. 9 287–300
[4] Caldas L G and Norford L K 2002 A design optimisation tool based on a genetic algorithm Autom. Constr. 11 173–84
[5] Wang W, Zmeureanu R and Rivard H 2005 Applying multi-objective genetic algorithms in green building design optimisation Build. Environ. 40 1512–25
Magnier L and Haghighat F 2010 Multiobjective optimisation of building design using TRNSYS simulations, genetic algorithm, and Artificial Neural Network Build. Environ. 45 739–46

Turrin M, Von Buelow P and Stouffs R 2011 Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms Adv. Eng. Informatics 25 656–75

Newton D 2020 Deep Generative Learning for the Generation and Analysis of Architectural Plans with Small Datasets eCAADe SIGraDi 2 21–8

Uzun C, Çolakoğlu M B and İnceoğlu A 2020 GAN as a generative architectural plan layout tool: A case study for training DCGAN with palladian plans and evaluation of DCGAN outputs A/Z ITU J. Fac. Archit. 17 185–98

Newton D 2019 Generative Deep Learning in Architectural Design Technol. Archit. Des. 3 176–89

Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2014 Generative adversarial networks Commun. ACM 63 139–44

Hoggarth P A, Innes C R H, Dalrymple-Alford J C and Jones R D 2015 arXiv2014_Conditional GAN_Conditional generative adversarial nets Accid. Anal. Prev. 77 29–34

Odena A 2016 Semi-supervised Learning with Generative Adversarial Networks arXiv

Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A and Chen X 2016 Improved techniques for training GANs Adv. Neural Inf. Process. Syst. 22 34–42

Radford A, Metz L and Chintala S 2016 Unsupervised representation learning with deep convolutional generative adversarial networks 4th Int. Conf. Learn. Represent. ICLR 2016 - Conf. Track Proc. 1–16