



OPEN

Interactive design generation and optimization from generative adversarial networks in spatial computing

Xiaochen Hu[✉], Cun Lin, Tianyi Chen & Weibo Chen

This paper focuses on exploring the application possibilities and optimization problems of Generative Adversarial Networks (GANs) in spatial computing to improve design efficiency and creativity and achieve a more intelligent design process. A method for icon generation is proposed, and a basic architecture for icon generation is constructed. A system with generation and optimization capabilities is constructed to meet various requirements in spatial design by introducing the concept of interactive design and the characteristics of requirement conditions. Next, the generated icons can effectively maintain diversity and innovation while meeting the conditional features by integrating multi-feature recognition modules into the discriminator and optimizing the structure of conditional features. The experiment uses publicly available icon datasets, including LLD-Icon and Icons-50. The icon shape generated by the model proposed here is more prominent, and the color of colored icons can be more finely controlled. The Inception Score (IS) values under different models are compared, and it is found that the IS value of the proposed model is 7.05, which is higher than that of other GAN models. The multi-feature icon generation model based on Auxiliary Classifier GANs performs well in presenting multiple feature representations of icons. After introducing multi-feature recognition modules into the network model, the peak error of the recognition network is only 2.000 in the initial stage, while the initial error of the ordinary GAN without multi-feature recognition modules is as high as 5.000. It indicates that the improved model effectively helps the discriminative network recognize the core information of icon images more quickly. The research results provide a reference basis for achieving more efficient and innovative interactive space design.

Keywords Spatial calculation, Generative adversarial networks, Interactive design generation, Auxiliary classifier

In modern design, exploring methods for intelligent design processes has become increasingly important to improve design efficiency and creativity. Designers hope to utilize computer-aided technology to achieve more efficient and innovative design processes through automation and intelligence. Generative Adversarial Networks (GANs), as a powerful Deep Learning (DL) technology, have shown astonishing innovation potential in multiple fields^{1,2}. Lv and Qiao³ pointed out that cognitive computing did not mean that computers replaced human thinking but rather made it an effective auxiliary tool for people to recognize and process large-scale data. How to fully utilize GANs to achieve interactive design generation and optimization to meet the constantly evolving design needs has become a hot research topic. As a creative field, spatial design requires not only meeting functional needs but also considering emotions, aesthetics, and user experience. However, traditional design processes are limited by manual operations and experience accumulation, making it difficult to efficiently implement complex design tasks. The emergence of GANs provides new possibilities for solving this problem. Through the game process of generators and discriminators, GAN can generate data samples with high fidelity, bringing revolutionary changes in design⁴⁻⁶.

In the past, the creativity and innovation of design mainly relied on the experience and intuition of designers, but this traditional approach often limited the diversity and efficiency of design. With the rise of data-driven methods, the role of computers in design has gradually become prominent^{7,8}. GAN is an adversarial framework composed of generators and discriminators, which can generate realistic data by allowing two neural networks to

School of Design and Innovation, China Academy of Art, Hangzhou 310000, Zhejiang, China. ✉email: huxiaochen@caa.edu.cn

play games with each other. This technology has achieved remarkable success in fields, such as image generation and speech synthesis, and its introduction into the design field has enormous potential. Spatial computing, as an important branch of the design field, covers multiple aspects, such as architecture, interior design, and urban planning. How to apply GANs in spatial design to achieve more innovative and diverse design generation and optimization has become a fascinating topic^{9–11}.

In the application of various generative models, images, as a common data form with high information density, exhibit a wide range of possibilities. The transformation from landscape painting to oil painting, horses to zebras, and youth to elders and even the variability of facial expressions, the repair of damaged images, the improvement of image resolution, and the innovation of text-generated images have all enriched the application field of image generation¹². These practical applications provide a multi-dimensional learning perspective, and image generation has become a crucial direction in generative models. With the advancement of artificial intelligence technology, the concept of automatically generating icons has gradually become a reality. Users can obtain preliminary ideas for various icons by inputting the required icon element features into the icon generation model. Then, users can select satisfactory icon styles and request designers to further refine them to ultimately create a satisfactory icon design.

This paper proposes an icon generation method based on GANs to meet the needs of spatial design. In this basic architecture, the generator and discriminator play games with each other to achieve high-quality icon generation. The concept of interactive design and requirement condition features are introduced to better adapt to the particularity of spatial design, and a system with generation and optimization capabilities is constructed. This paper introduces multi-feature recognition modules in the discriminator to maintain the diversity and innovation of the generated icons while meeting the conditional features. It is expected to effectively improve the diversity and fidelity of icon generation by optimizing the structure of conditional features.

Literature review

At present, traditional image generation models typically use methods, such as image combination, cropping, color adjustment, and angle change, to create. With the advancement of DL neural network technology, the current research on generating models has focused on using neural networks to fit the distribution of raw image data. Then, a brand new image is generated based on this distribution. With the rapid development of DL technology, GAN has made significant research progress in interactive design generation and optimization in spatial computing. Gan et al.¹³ proposed an interactive urban layout generation method based on GAN. They used generators to generate candidate city layouts, and users could interact with the generation process by adjusting sketches in real-time. The discriminator evaluated the similarity between the generated layout and the real city layout, guiding the generation process and achieving collaborative design generation between users and the system. Niu et al.¹⁴ explored the method of using GAN to automatically generate product designs. The research team designed a generator network that used sketches as input to generate various possible product designs. The discriminator evaluated based on the existing product database, guiding the generator to generate more innovative and practical designs.

Hu et al.¹⁵ introduced a Three-Dimensional (3D) spatial data augmentation method based on auxiliary GAN. Researchers proposed SpatialGAN, which generated diverse 3D spatial data through generators to enhance the training dataset and improve the performance of DL-based spatial computing models. Dan et al.¹⁶ explored how to use deep generation models for interactive design space exploration. Researchers developed an interactive interface that allowed users to explore the design space while generating diverse design suggestions based on GAN to help users better understand and define design goals. Tang et al.¹⁷ conducted a comparative review on the application of different deep generation models in urban design. It covered the research progress of different models, such as variational autoencoders, GANs, and generative flows. Researchers evaluated the effectiveness and limitations of these methods in achieving interactive design generation and optimization from multiple perspectives.

Although existing methods have achieved interaction between user and generator, there are challenges in integrating user feedback. How to effectively integrate user feedback into the design generation and optimization process still requires further exploration. This paper focuses on how to more effectively integrate user feedback into the design generation and optimization process. The generator can more accurately understand the user's intentions and continuously optimize the generated design based on user feedback by developing an intelligent feedback mechanism.

Materials and methods

Icon generation model based on GANs

The GAN is a network structure composed of generators and discriminators. The generator aims to generate samples similar to real data, while the discriminator aims to distinguish between the generated samples and real data. Through continuous adversarial learning, the generator gradually improves the fidelity of the generated samples, enabling them to be confused with the real. The core idea of GAN is to train the generated model through adversarial training of generators and discriminators^{18–20}. The structural diagram of GAN is shown in Fig. 1. The left side of Fig. 1 shows the generated network G, with an input of a set of random noise Z and an output of new data G(z), aiming to make its distribution as close as possible to the real data distribution. The architecture of the generated network can be seen as the reverse structure of discriminant networks, creating new data with or without targeting through multi-layer deconvolution calculations. The successful quality of generating a network depends on the way it optimizes its structure and parameters. Training and sampling data are conducted in a generative network to measure the skill level of researchers in analyzing and calculating high-dimensional probability distributions.

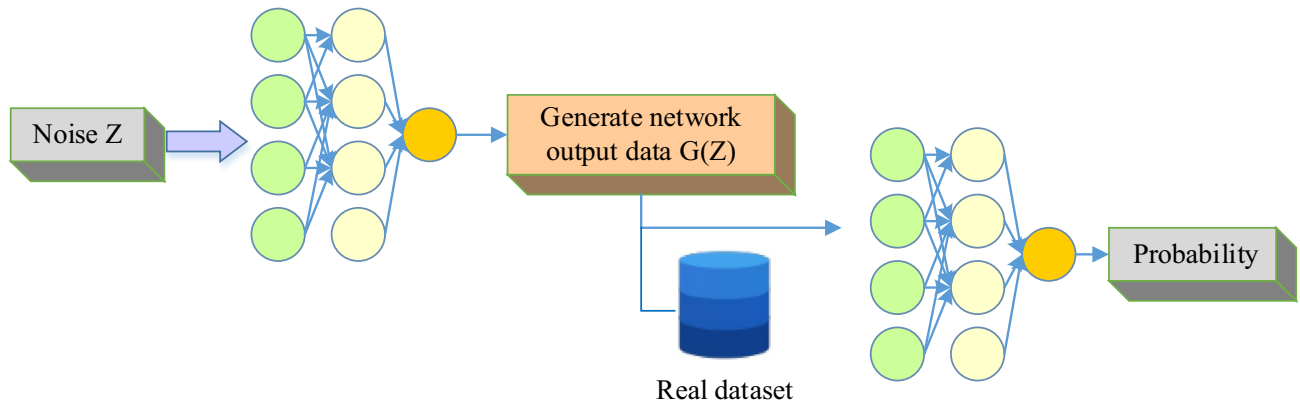


Figure 1. Structural schematic diagram of GAN.

The generator is a key component of icon GAN (IconGAN), whose task is to accept a random noise vector (usually represented as Z) and transform it into a new icon as similar as possible to the real icon. The generator adopts a multi-layer deconvolution network structure, gradually increasing the details and features of the icon to achieve icon generation. The optimization goal of the generator is to make the generated icons realistic enough to mislead the discriminator.

The discriminator is another important component of IconGAN, whose task is to evaluate the similarity between the generated icon and the real icon. The input of the discriminator can be a real icon or an icon generated by the generator, with the goal of distinguishing between the two. The discriminator gradually learns how to distinguish between generated icons and real icons by judging between generated icons and real icons^{21,22}. The optimization goal of discriminators is to improve their judgment ability to better distinguish the authenticity of generated icons. The training process of IconGAN is based on adversarial learning between the generator and discriminator. In each training iteration, the generator generates a set of icons, and the discriminator judges between these generated and actual icons. The goal of the generator is to deceive the discriminator by generating icons, making it difficult to distinguish between generated and real icons. The discriminator aims to accurately determine which icons are generated and which are real. Through continuous adversarial learning, the generator gradually improves the quality of generated icons, making them more realistic. The basic framework of IconGAN enables its widespread application in icon design^{23–25}. IconGAN can generate high-quality icons similar to real icons by optimizing the structure and parameters of the generator and discriminator. The generated icons can be customized according to specific design requirements to meet the needs of different application scenarios, such as mobile applications, website design, and brand identification. Figure 2 shows the basic framework of icon GANs.

Identification module incorporating multiple features

In traditional GAN, discriminators usually only rely on global features to determine the authenticity of the generated samples. This can easily lead to the generated samples that are not realistic enough in some details, and unstable phenomena, such as pattern collapse and pattern collapse, can also occur during the training process^{26,27}. Multi-scale discriminator is an improved approach that can capture image information more comprehensively by introducing discriminators of different scales. In the image generation task, discriminators can be trained on the original image and the image with reduced size simultaneously^{28–30}. This multi-scale discriminator can capture features at different levels from details to the global, improving the authenticity and diversity of generated samples.

The multi-level feature discriminator proposed here is a structure that designs the discriminator as multiple sub-discriminators. Each sub-discriminator is responsible for different levels of features, such as low-level textures and high-level semantics. This design enables the discriminator to analyze images at multiple levels of abstraction, thereby improving the judgment ability and diversity of generated samples. A multimodal discriminator is a method that integrates multiple features and considers different aspects of an image, such as appearance and semantic information. A multimodal discriminator is a model component designed to handle multiple data modalities or features. In DL, this may involve multiple neural network branches, each specialized in processing a specific data type (such as images, text, and sound). The outputs of these branches may be integrated into a common representation or used for different tasks, such as classification and generation. The multimodal discriminator helps the model understand and utilize the relationships among different data types, thereby improving the model's performance. The model can simultaneously focus on multiple aspects of the image by introducing a multimodal discriminator, thereby more accurately evaluating the authenticity of the generated samples. This helps generate more creative and diverse samples.

The improved Auxiliary Classifier GAN (ACGAN) has significant differences from the auxiliary GAN. In ACGAN, the generator still receives conditional features, but the design of the discriminator differs from traditional methods. In the discriminator, conditional features are no longer directly input but are added to the output through the conditional feature recognition module. This makes the task of the discriminator clearer, with the goal of checking whether the input image meets the conditional features expected by the generator. Therefore, in ACGAN, the generator is responsible for generating images that meet the conditional features, while the discriminator is responsible for verifying whether the images meet the conditions. The model structure is given

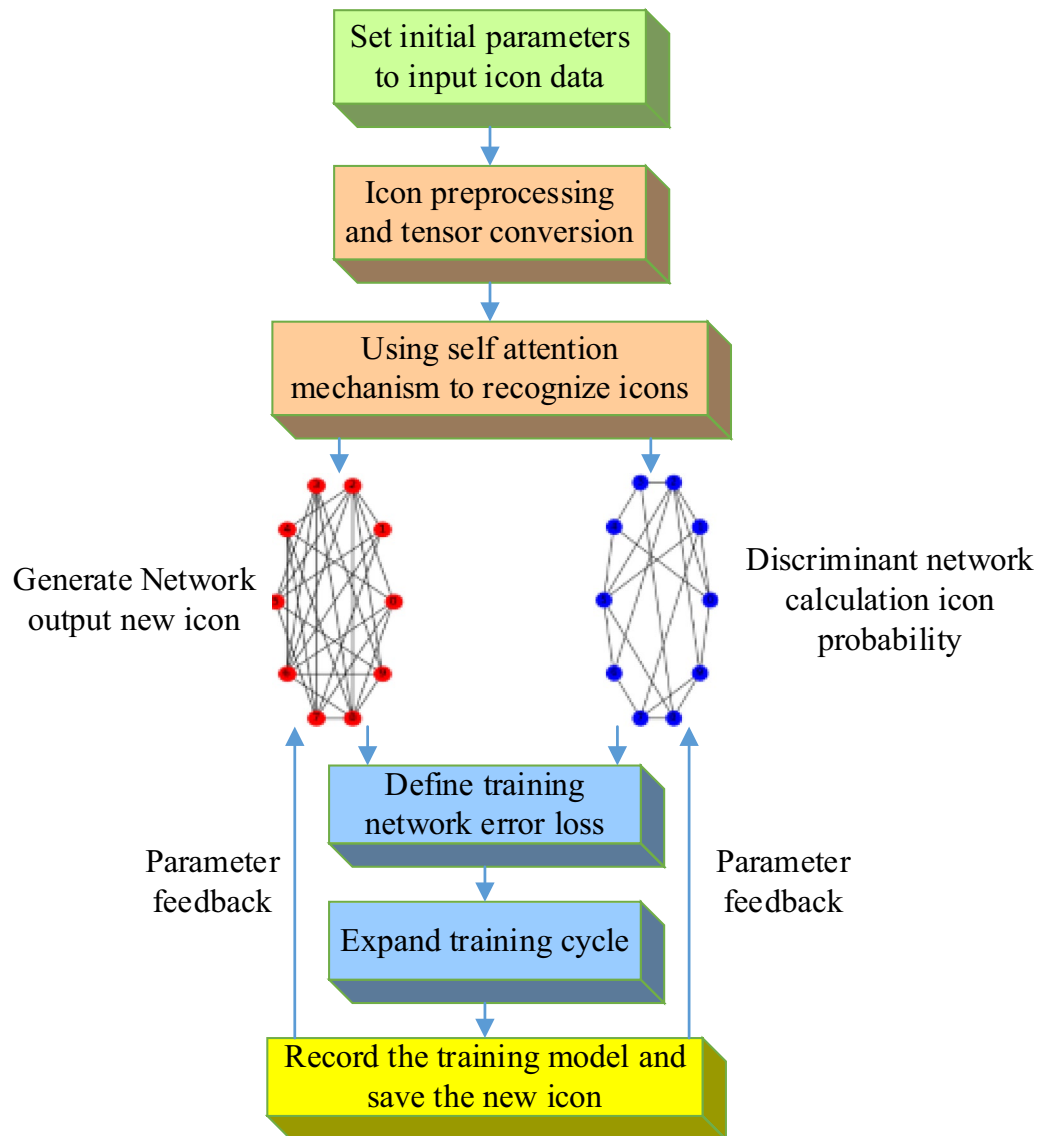


Figure 2. Basic framework for icon GANs.

in Fig. 3. This method more effectively captures important conditional features of the image, such as color and shape, making the generated image closer to reality.

With the Wasserstein GAN (WGAN) model, the stability and generation efficiency of ACGAN can be further improved^{31,32}. The distance among distributions is measured using Wasserstein distance. It introduces color condition features and the shape condition feature s . In this way, the Auxiliary Classifier Wasserstein GAN (ACWGAN) model is formed, and the loss functions of the generator and discriminator are:

$$L_D = E_{z \sim p_z(z)} [f(G|z(c, s)|c, s)] - E_{x \sim p_{r|c, s}(x|c, s)} f(x|c, s) \quad (1)$$

$$L_G = -E_{z \sim p_z(z)} [f(G|z(c, s)|c, s)] \quad (2)$$

The function f follows the constraint of Lipschitz constant, and the parameters of this function are obtained through neural network training. To meet the limit of Lipschitz constant, the parameter ω is constrained to a pre-set cutoff parameter h during each round of parameter updates. In addition, the conditional features of the model are composed of color condition feature c and shape condition feature s , which can collectively affect the generation of icons. If necessary, further features can be added to make the generated icons more targeted.

After selecting and constructing the icon generation model, it is necessary to clarify the training method and evaluation strategy of the model. Here, the initial ACGAN is improved, and a multi-class GAN model is proposed. The algorithm process is as follows. Batch training is adopted for iterative training, which involves extracting data from the dataset to form batches for training. The specific training process is shown in Table 1.

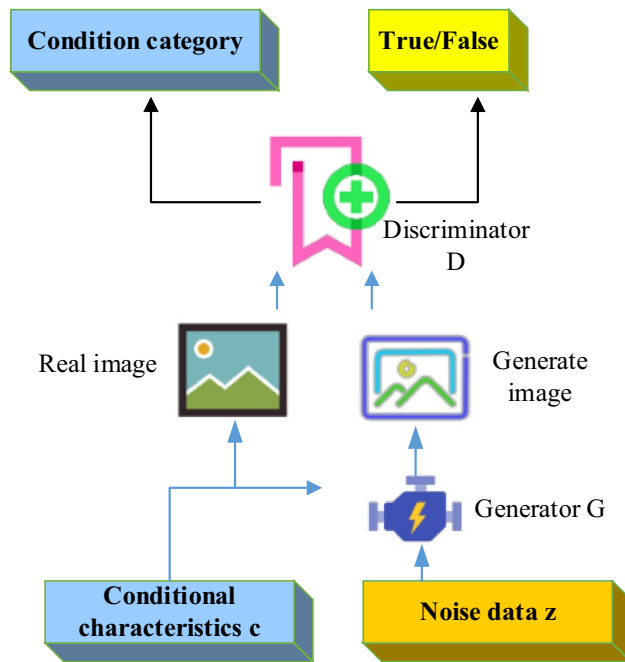


Figure 3. Icon generation model based on ACGAN.

Input: learning rate α , cutoff parameter h , batch parameter m , rounds of identification of one generation n_{critic} , z_{size} noise length, w_0 initial discriminator parameter, θ initial generator parameters, shape condition features s , color condition feature c
1. Initialization parameters
2. while
3. for $t=0, \dots, n_{critic}$ do // discriminator
4. Collect a batch of data from a real dataset $\{x^{(i)}\}_{i=1}^m \sim P_r$
5. Generate a batch of data according to distribution $\{z^{(i)}\}_{i=1}^m \sim P_r$
6. Collect a batch of data from the color and shape condition label y , which are $\{c^{(i)}\}_{i=1}^m, \{s^{(i)}\}_{i=1}^m$
7. update $\omega \leftarrow \omega + \alpha RMSProp(\omega, g_\omega)$
8. Control parameters within the specified range $\omega \leftarrow clip(\omega, -h, h)$
9. end for
10. Generate a batch of data according to distribution $\{z^{(i)}\}_{i=1}^m \sim P(z)$ // generator
11. update g_θ
12. update θ
13. end while

Table 1. Training process based on ACWGAN model.

Construction of an icon generation network for conditionally assisted improvement

Conditional inputs, usually specified features or parameters, are introduced into the icon generation network to customize the generated icons. These conditions can be color, shape, size, and style. Conditional input guides the generator to generate icons with specific styles or features, achieving customized generation. The appropriate generator and discriminator network architecture is selected to adapt to conditional inputs. The generator needs to combine conditional input with random noise to generate icons that match the conditions. The discriminator needs to determine whether the generated icon meets the conditions^{33,34}.

In the generation network, the previously obtained label needs to be converted into a label tensor label (label_tensor) suitable for matrix calculation, which is simultaneously passed in a flat layer and an embedding layer to convert the two-dimensional output matrix into a one-dimensional vector. Subsequently, the number of labels (num_classes) is combined. Then, it is combined with noise through the superposition function to generate input. Finally, the input is merged with the label tensor (label_tensor) to form a new input for the generated network. Therefore, the conditional variable C is regarded as the limiting design concept in icon design, which brings constraints to the design direction and output results of the training network.

Relatively speaking, the processing of discriminative networks is simpler. When processing inputs and conditions, operations are carried out through a fully connected layer to ensure that the output dimension is consistent

with the number of labels. Next, after a single-layer normalization operation, the results are transmitted to the training process. In the formal training phase, the labels are first randomly sampled to obtain the sampled labels (sampled_labels) and generate new icons (gen_imgs). d_loss_real calculates the loss between the input imgs and the original label (img_labels), using the valid values (valids) as the input. d_loss_new evaluates the loss between the newly generated icon (gen_imgs) and the sampled label (sampled_labels), and the input is a false value (fake).

Experimental design

During the model training process, a series of key parameter settings are involved, which play a crucial role in the performance and effectiveness of the model. The optimization function settings of the WGAN model are referenced, and RMSprop is used as the optimization function. The learning rate of the neural network is set to 0.00005, while the batch size of the data is set to 32. Besides, the parameter update process of the neural network refers to the cutoff parameter of the WGAN model, set to 0.01. In the specific construction of the model, the activation function of the generator adopts Rectified Linear Unit (ReLU), while the output of the generator follows the Tanh activation function in the WGAN model. To better train the discriminator, Leaky ReLU is used as the activation function in its convolutional layer, and the parameter value is set to 0.2.

The specific network parameter settings are shown in Table 2, which aim to achieve better convergence and generation effect during the training process. The selection of these parameters has undergone sufficient experiments and optimization to ensure that the improved model can achieve satisfactory performance in path planning problems.

The Python-based machine learning framework Tensorflow is adopted to build the model, and the testing platform for building the model is configured accordingly. The configuration of the testing platform includes a 3.9 GHz Intel Core i7 CPU, Nvidia RTX-2070 8 GB GPU, and 32 GB of memory.

Two different icon datasets are used in the experiment, namely Icons-50 and LLD-Icon. The LLD-Icon dataset is a large-scale icon dataset created by researchers, such as Alexander Sage. Unlike traditional icon datasets, the characteristic of the LLD-Icon dataset is that it is collected from the internet and contains many real-world icon samples. This dataset contains 486,377 icons, each with an image size of 32×32 pixels with 3 color channels. This means that the image of each icon is an RGB color image. The Icons-50 dataset is another commonly used icon dataset, mainly used for icon recognition and image classification tasks. The Icons-50 dataset is manually annotated, and each icon is assigned to a predefined 50 categories. Each category represents a different type of icon, and each icon image is also a 32×32 pixel RGB color image.

After obtaining the dataset, it must be preprocessed to adapt to the input criteria of the model. Firstly, the size of image data is standardized to 28×28 pixels. The data is divided into two groups: color and black and white to conduct a more comprehensive experiment. In the black and white icon group, the OpenCV image processing library is used to convert color images into black and white images. The Scikit-learn scientific computing library is used to divide the dataset into training and testing sets, with the testing set accounting for 20% of the total sample size.

Results and discussion

Analysis of icon generation effect

The ACWGAN proposed here is validated on the Icons-50 and LLD-Icon datasets for the generation of black and white and color icons. The specific results are revealed in Figs. 4, 5, 6 and 7. The experimental results show that the conditional GAN model with conditional constraints has stronger directionality based on GAN models. Especially in generating black and white icons, the model can more prominently highlight shape features. The generated bicycle icon can be clearly seen, which fully demonstrates the important role of conditional features in icon generation. Compared to using only GAN models, the improved model incorporating conditional features is more capable of generating icon styles with more target features based on specific conditions. This also indirectly verifies the effectiveness of the conditional GAN model in icon generation tasks.

Network layer		Convolutional kernel	Step	Depth	Dropout	Activation function	Normalization
Input:G(z)-100	Dense	-	-	128*7*7	0	-	ReLU
	Reshape	-	-	7,7,128	0	-	-
	UpSampling	-	-	-	0	-	-
	Convolution	4×4	1×1	128	0	Yes	ReLU
	UpSampling	-	-	-	0	-	-
	Convolution	4×4	1×1	64	-	Yes	ReLU
	Convolution	4×4	1×1	3	-	-	Tanh
D(x,z)-28*28*1	Convolution	3×3	2×2	16	0.25	Yes	-Leaky ReLU
	Convolution	3×3	2×2	32	0.25	Yes	Leaky ReLU
	Convolution	3×3	2×2	64	0.25	Yes	Leaky ReLU
	Convolution	3×3	1×1	128	0.25	Yes	Leaky ReLU
	Flatten	-	-	-	-	-	-
	Dense	-	-	1	-	-	-

Table 2. Parameter settings based on ACWGAN.

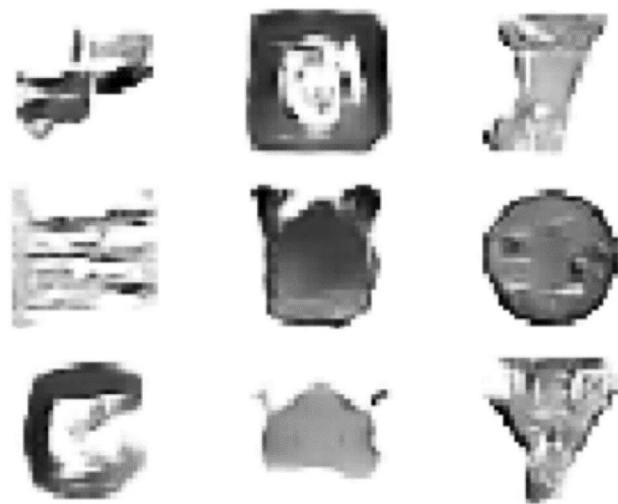


Figure 4. LLD-Icon black and white icon generation rendering based on ACWGAN.

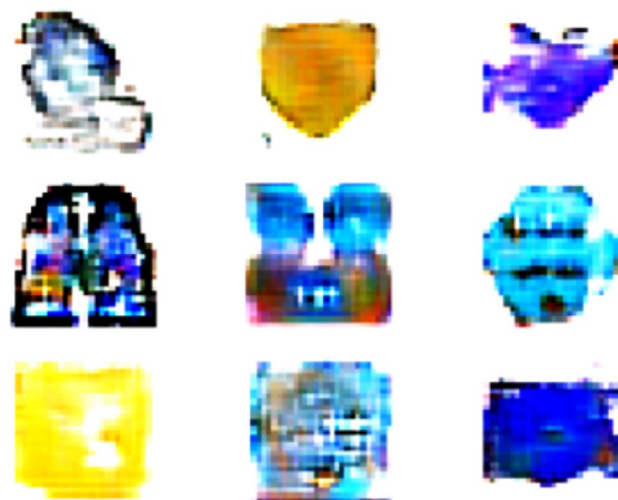


Figure 5. LLD-Icon color icon generation rendering based on ACWGAN.



Figure 6. Icons-50 black and white icon generation rendering based on ACWGAN.

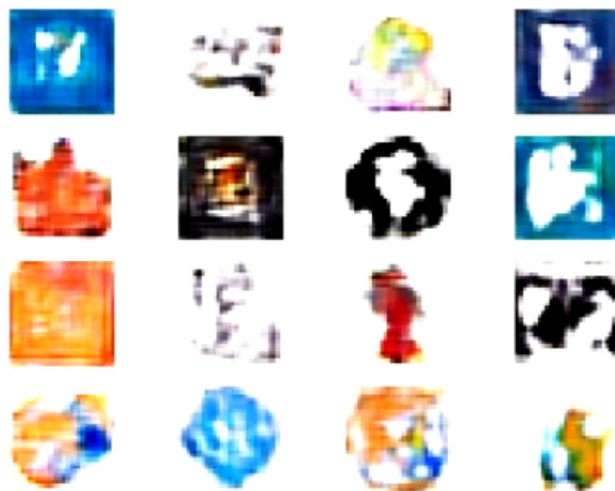


Figure 7. Icons-50 color icon generation rendering based on ACWGAN.

Comparison of Inception Score (IS) and Fréchet Inception Distance (FID) values for different generation models

Figures 8 and 9 show the dynamic changes in FID and IS values of the GAN model, WGAN model, ACGAN model, and the proposed ACWGAN model on the Icons-50 dataset. Generally speaking, a smaller FID value indicates that the distribution of generated samples is closer to the distribution of real samples. Additionally, the higher the IS value, the higher the authenticity score of the generated samples, and the higher the clarity of the generated samples. These two evaluation indicators jointly reveal the superiority of the generated model, and the generated samples can better simulate the distribution of real data and have higher performance in terms of quality and clarity.

After a series of iterations, the FID value of the ACWGAN model rapidly decreases, while the IS value gradually increases to 7.05, resulting in the generated image gradually becoming clearer. As the iteration continues, the mutual confrontation between the generative model and the discriminative model leads to a continuous improvement in the quality of the generated image. At around 100,000 iterations, the ACWGAN model tends to stabilize, while other comparative models are slightly unstable and even experience model crashes. This leads to a sharp increase in FID values and a sharp decrease in IS values. Moreover, after introducing multi-feature recognition modules into the model, the peak error of the recognition network is only 2.000 in the initial stage, while the initial error of the ordinary GAN without multi-feature recognition modules is as high as 5.000. It indicates that the improved model effectively helps the recognition network recognize the core information of the icon image faster.

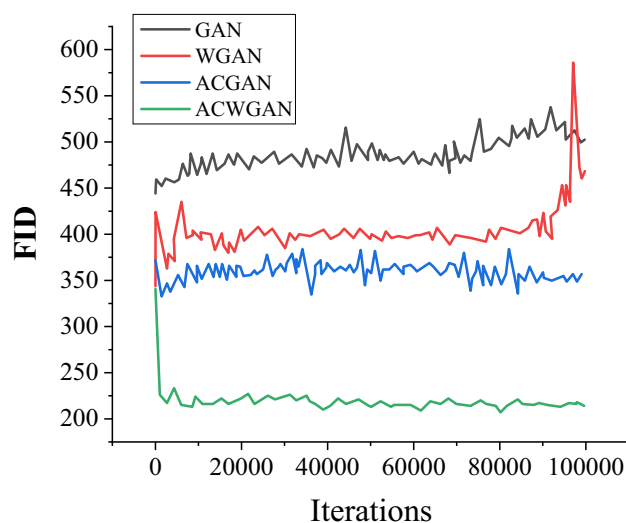


Figure 8. Comparison of FID values for different generation models.

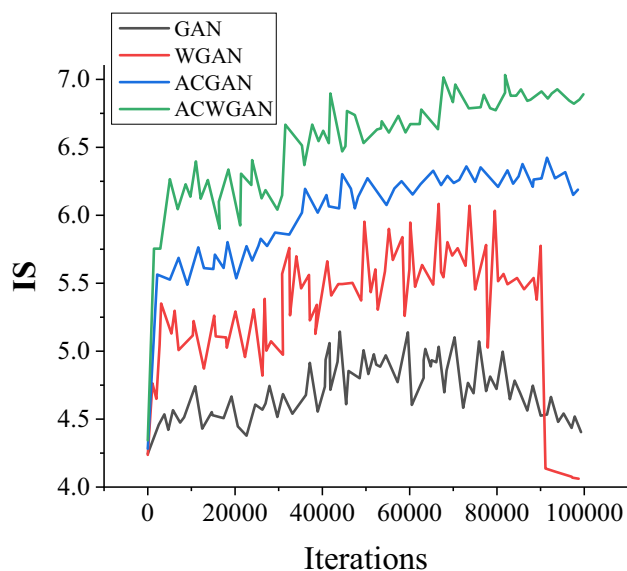


Figure 9. Comparison of IS values for different generation models.

Conclusion

This paper explores the application potential and optimization problems of GANs in spatial computing and proposes an interactive design generation and optimization method based on multiple features. It constructs a basic architecture by establishing icons, introduces the concept of interactive design and requirement condition features, and constructs a system with generation and optimization capabilities, aiming to meet the diverse needs of different spatial design fields. The generated icons have successfully achieved greater diversity and innovation while maintaining conditional features by integrating multi-feature recognition modules into the discriminator. The experimental results show that the multi-feature icon generation model ACWGAN proposed here performs well in shape highlighting and color fine control and demonstrates its potential application in spatial design.

Although this paper introduces conditional features and multi-feature recognition modules to optimize the generation process, there is still some optimization space in practical applications. For example, how to better determine and design conditional features and how to select and fuse multiple features are issues that require further research. In addition, more GAN structures can be explored, such as variational autoencoder GANs, to further improve the performance of the generation model.

Data availability

All data generated or analyzed during this study are included in this published article [and its supplementary information files].

Received: 9 October 2023; Accepted: 16 February 2024

Published online: 02 March 2024

References

1. Wu, A. N., Stouffs, R. & Biljecki, F. Generative Adversarial Networks in the built environment: A comprehensive review of the application of GANs across data types and scales. *Build. Environ.* **223**, 109477 (2022).
2. Shu, D. *et al.* 3d design using generative adversarial networks and physics-based validation. *J. Mech. Des.* **142**(7), 071701 (2020).
3. Lv, Z. & Qiao, L. Deep belief network and linear perceptron based cognitive computing for collaborative robots. *Appl. Soft Comput.* **92**, 106300 (2020).
4. Huang, C. *et al.* Accelerated environmental performance-driven urban design with generative adversarial network. *Build. Environ.* **224**, 109575 (2022).
5. Gan, Y. *et al.* Integrating aesthetic and emotional preferences in social robot design: An affective design approach with Kansei Engineering and Deep Convolutional Generative Adversarial Network. *Int. J. Ind. Ergon.* **83**, 103128 (2021).
6. Jabbar, A., Li, X. & Omar, B. A survey on generative adversarial networks: Variants, applications, and training. *ACM Comput. Surv.* **54**(8), 1–49 (2021).
7. Hughes, R. T., Zhu, L. & Bednarz, T. Generative adversarial networks-enabled human-artificial intelligence collaborative applications for creative and design industries: A systematic review of current approaches and trends. *Front. Artif. Intell.* **4**, 604234 (2021).
8. Wang, B. *et al.* Hyperspectral imagery spatial super-resolution using generative adversarial network. *IEEE Trans. Comput. Imaging* **7**, 948–960 (2021).
9. Ye, X., Du, J. & Ye, Y. MasterplanGAN: Facilitating the smart rendering of urban master plans via generative adversarial networks. *Environ. Plan. B* **49**(3), 794–814 (2022).
10. Hu, Z. *et al.* Sketch2VF: Sketch-based flow design with conditional generative adversarial network. *Comput. Animat. Virtual Worlds* **30**(3–4), e1889 (2019).
11. Long, T. *et al.* Constrained crystals deep convolutional generative adversarial network for the inverse design of crystal structures. *npj Comput. Mater.* **7**(1), 66 (2021).

12. Dong, W., Chen, X. & Yang, Q. Data-driven scenario generation of renewable energy production based on controllable generative adversarial networks with interpretability. *Appl. Energy* **308**, 118387 (2022).
13. Gan, P. G. *et al.* Thermal properties of nanocellulose-reinforced composites: A review. *J. Appl. Polym. Sci.* **137**(11), 48544 (2020).
14. Niu, S. *et al.* Defect image sample generation with GAN for improving defect recognition. *IEEE Trans. Autom. Sci. Eng.* **17**(3), 1611–1622 (2020).
15. Hu, Y. *et al.* Towards a more realistic and detailed deep-learning-based radar echo extrapolation method. *Remote Sens.* **14**(1), 24 (2021).
16. Dan, Y. *et al.* Generative adversarial networks (GAN) based efficient sampling of chemical composition space for inverse design of inorganic materials. *npj Comput. Mater.* **6**(1), 84 (2020).
17. Tang, T. W. *et al.* Anomaly detection neural network with dual auto-encoders GAN and its industrial inspection applications. *Sensors* **20**(12), 3336 (2020).
18. Kammoun, A. *et al.* Generative adversarial networks for face generation: A survey. *ACM Comput. Surv.* **55**(5), 1–37 (2022).
19. Bharti, V., Biswas, B. & Shukla, K. K. EMOCGAN: a novel evolutionary multiobjective cyclic generative adversarial network and its application to unpaired image translation. *Neural Comput. Appl.* **34**(24), 21433–21447 (2022).
20. Li, L. *et al.* Text to realistic image generation with attentional concatenation generative adversarial networks. *Discret. Dyn. Nat. Soc.* **2020**, 1–10 (2020).
21. Qian, W., Xu, Y. & Li, H. A self-sparse generative adversarial network for autonomous early-stage design of architectural sketches. *Comput.-Aided Civ. Infrastruct. Eng.* **37**(5), 612–628 (2022).
22. Zhou, K., Diehl, E. & Tang, J. Deep convolutional generative adversarial network with semi-supervised learning enabled physics elucidation for extended gear fault diagnosis under data limitations. *Mech. Syst. Signal Process.* **185**, 109772 (2023).
23. Liu, W. *et al.* A hybrid quantum-classical conditional generative adversarial network algorithm for human-centered paradigm in cloud. *EURASIP J. Wirel. Commun. Netw.* **2021**(1), 1–17 (2021).
24. Park, S. W. *et al.* Review on generative adversarial networks: focusing on computer vision and its applications. *Electronics* **10**(10), 1216 (2021).
25. Amimeur, T. *et al.* Designing feature-controlled humanoid antibody discovery libraries using generative adversarial networks. *BioRxiv* **109**, 74 (2020).
26. Wang, C. *et al.* Framework of nacelle inverse design method based on improved generative adversarial networks. *Aerosp. Sci. Technol.* **121**, 107365 (2022).
27. Liao, W. *et al.* Automated structural design of shear wall residential buildings using generative adversarial networks. *Autom. Constr.* **132**, 103931 (2021).
28. Dong, Y. & Ren, F. Multi-reservoirs EEG signal feature sensing and recognition method based on generative adversarial networks. *Comput. Commun.* **164**, 177–184 (2020).
29. Zhang, H. *et al.* MASG-GAN: A multi-view attention superpixel-guided generative adversarial network for efficient and simultaneous histopathology image segmentation and classification. *Neurocomputing* **463**, 275–291 (2021).
30. Hayashi, H., Abe, K. & Uchida, S. GlyphGAN: Style-consistent font generation based on generative adversarial networks. *Knowl.-Based Syst.* **186**, 104927 (2019).
31. Yuan, C. & Moghaddam, M. Attribute-aware generative design with generative adversarial networks. *IEEE Access* **8**, 190710–190721 (2020).
32. Karimi, M. *et al.* De novo protein design for novel folds using guided conditional Wasserstein generative adversarial networks. *J. Chem. Inf. Model.* **60**(12), 5667–5681 (2020).
33. Nguyen, P. C. H. *et al.* Synthesizing controlled microstructures of porous media using generative adversarial networks and reinforcement learning. *Sci. Rep.* **12**(1), 9034 (2022).
34. Liu, M. Y. *et al.* Generative adversarial networks for image and video synthesis: Algorithms and applications. *Proc. IEEE* **109**(5), 839–862 (2021).

Acknowledgements

New Liberal Arts Research and Reform practice Project of the Ministry of Education. Training Innovation and practice of compound talents in New Arts and Arts majors 2021100046.

Author contributions

X.H., C.L., and T.C. contributed to conception and design of the study. W.C. organized the database. X.H. performed the statistical analysis. C.L. wrote the first draft of the manuscript. T.C., W.C., and X.H. wrote sections of the manuscript. All authors contributed to manuscript revision, read, and approved the submitted version.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-024-54783-6>.

Correspondence and requests for materials should be addressed to X.H.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024