International Journal for Housing Science and Its Applications
Publish August 10, 2025. Volume 46, Issue 4 Pages 1888-1902

https://doi.org/10.70517/ijhsa464163

Research on Intelligent Computational Models for Building Layout and Functional Zoning Optimization in Urban Spatial Planning

Min Ma[1,*]
[1] Southeast University Architectural Design and Research Institute Co., Ltd., Nanjing, Jiangsu, 210096, China
Corresponding authors: (e-mail: mmmm_1242a@163.com).

Abstract This paper proposes an automatic layout method for high-rise residential buildings based on deep deterministic policy gradient descent, combining deep reinforcement learning techniques. Building regulations such as land use boundaries, sunlight requirements, and building spacing—which are key considerations in residential area layout—are extracted and formulated into computer-understandable rules. Multiple constraints and optimization objectives are unified within a single framework. Subsequently, based on the actual scenario, the state space, action space, and reward function are designed to perform automatic optimization of building layout. To efficiently generate optimal layout schemes for residential areas, a generation process based on conditional generative adversarial networks (CGAN) is designed to generate building functional zoning schemes and conduct validation and evaluation. The results indicate that the automatically generated urban spatial building layout design diagrams under this algorithm comply with regulations. Furthermore, this study found that as the amount of data increases, the number of times the model achieves optimal training results decreases significantly. For example, when the data volume is 800, the number of training iterations required for the model to achieve optimal results is reduced by over 50% compared to when the data volume is 200, and the accuracy of the discriminator is also higher and more stable under these conditions. This indicates that the building layout schemes designed in this study meet planning requirements and provide an efficient and intelligent solution for urban spatial planning.

Index Terms deep reinforcement learning, CGAN, building layout, intelligent computational model

I. Introduction

Urban spatial planning refers to the systematic layout and design of urban space to achieve the goals of sustainable urban development and improved quality of life for residents [1], [2]. Among these, building layout and functional zoning are key components of urban planning [3].

Building layout refers to the arrangement and distribution of buildings across different areas of a city. Its purpose is to optimize the efficiency of space utilization while maintaining the overall structure and macro-level layout of the city, thereby enhancing the quality of life for its inhabitants [4]-[7]. Building layout is primarily divided into central business districts, residential areas, industrial zones, and educational districts [8]. Functional zoning, on the other hand, involves dividing and managing the city based on different functional requirements [9]. Through reasonable functional zoning, the city's organizational structure and efficiency can be improved, and resources can be allocated more effectively [10]. Functional zoning is primarily divided into administrative districts and commercial zones, among others [11]. Traditional urban spatial planning is proposed by humans and suffers from issues such as low efficiency and strong subjectivity, which are detrimental to the sustainable development of urban planning [12], [13].

In recent years, artificial intelligence technology has seen rapid development, and numerous intelligent computing methods have been proposed, including genetic algorithms, evolutionary algorithms, heuristic algorithms, particle swarm algorithms, hybrid intelligent algorithms, immune algorithms, neural networks, and machine learning [14]-[17]. The development of these intelligent algorithms has played a crucial role in advancing the optimization of building layout and functional zoning in urban spatial planning, and intelligent computing-based urban planning optimization models have gradually become a hot topic in the field of urban planning [18]-[21]. Intelligent computing models can assess the impact of different building layout and functional zoning schemes on economic development, including industrial layout and commercial facilities, and can be used to evaluate the optimization potential of existing layouts and propose optimization recommendations [22]-[25].

Literature [26] proposes a design optimization tool supporting performance-based architectural design, the Multi-Objective Architectural Design Explorer (MADE), along with a design exploration strategy, demonstrating its effectiveness in aspects such as architectural spatial layout and energy performance. Literature [27] emphasizes

the important role of urban spatial form analysis and points out the shortcomings of traditional research. By simplifying urban building forms, extracting building form templates, and establishing an association between building forms and fractal dimensions, the study provides references for urban spatial form optimization. Literature [28] outlines computational methods for architectural spatial design and analyzes progress in areas such as user preferences and environmental data integration, emphasizing the shift from rule-based methods to data-driven approaches. Literature [29] proposes layout parameterization methods based on interdisciplinary architectural layout research and introduces various layout types, revealing that architectural layout has a significant influence on the average wind speed in the surrounding space. The above studies reveal that methods such as multi-objective architectural design explorers and layout parameterization can provide references for the rational layout of urban architecture by proposing their application in urban architectural layout.

Literature [30] points out the shortcomings of current urban functional zoning research and constructs an optimization framework for urban main functional zoning based on the “element-space-function” analytical logic, revealing that main functional zoning optimization methods provide differentiated strategies and guidance for future regional development. Literature [31] emphasizes the importance of constructing an analytical framework for optimizing urban spatial structure and function. It applies the minimum cumulative resistance model to delineate ecological suitability zones in Beijing, verifying that the proposed framework facilitates the optimization of urban spatial structure and function, and provides insights and technical support for the rational layout of urban spatial structure. Literature [32] points out the limitations of traditional urban zoning and spatial analysis methods, proposing a framework that combines spatial data analysis with advanced optimization algorithms to achieve efficient urban zoning. It verifies that this framework can provide accurate predictions and optimization solutions for urban spatial allocation. Literature [33] analyzes the similarities and differences between urban master planning and main functional zones based on the consistency of reality and objectives. The results indicate that the two share a degree of unity. From the perspectives of spatial boundary theory, functional positioning, and other aspects, the study validates the nature and content of main functional zones and urban master planning control zones, with the aim of promoting the scientific development of urban planning. Literature [34] examines the optimization of ecological service functions and planning control in land-use planning based on ecological protection and restoration. Considering the direct economic benefits of ecosystems, it constructs an ecological space evaluation scheme from an ecosystem service perspective, providing a reference for land-use planning. Literature [35] emphasizes the importance of reasonable division of urban functional zones for urban development. Using OpenStreetMap and point-of-interest data, combined with urban construction land classification standards, it identifies urban functional zones to reveal the impact of different urban areas on the urban thermal environment, with public service facility land having the greatest impact. The above studies demonstrate the importance of reasonable urban functional zoning for sustainable urban development. By proposing methods such as an optimized framework for urban main functional zoning and an analytical framework for urban spatial structure and functional optimization, these studies validate that such methods are conducive to achieving efficient urban functional zoning.

This paper first defines reinforcement learning and the Markov decision process, then introduces the overall framework for building automatic layout design, and provides constraint design, state space, and reward function design to construct a building automatic layout model that meets multiple constraints. Then, deep learning technology is applied to the automatic generation of residential area layout design schemes, proposing a design method for generating residential area functional zoning schemes based on CGAN. Considering the differences in solar radiation across regions, dataset samples are selected to analyze the effectiveness of building layout automatic generation based on the DDPG algorithm. Finally, the optimization effects of residential group layout and functional zones in urban spatial planning are analyzed.

II. Automatic layout model for urban buildings based on reinforcement learning

II. A. Reinforcement Learning

II. A. 1) Introduction to Reinforcement Learning

Reinforcement learning is a computational method in which machines interact with their environment to achieve specific goals. In the reinforcement learning framework, an agent is in a specific state and selects an action based on its current strategy. The environment responds to this action, causing a change in state and providing the agent with a reward signal. This reward signal is feedback on the action taken by the agent, aimed at guiding the agent on how to adjust its future behavior. The agent needs to continuously learn and adjust its strategy to improve the effectiveness of its decisions.

The core elements of reinforcement learning are as follows:

(1) State s is a representation variable generated based on the external environment. Depending on the actual situation, it may be a continuous or discrete variable, and the state space is S.

(2) Action a is a representation variable of the intelligent agent's behavior. Depending on the actual situation, it may be a continuous or discrete variable, and the action space is A.

(3) The primary role of the policy $\pi(a \mid s)$ is to analyze the current environmental conditions and make decisions regarding the subsequent action a;

(4) The state transition probability $p(s' \mid s,a)$ is the probability that the environment transitions to s' after action a is executed;

(5) The immediate reward $r(s,a,s')$ is a scalar function that represents the process of rewarding the environment based on the outcome of action a after it is executed.

The optimization objectives of reinforcement learning and supervised learning are similar, both aiming to optimize the expected value of a score under a given data distribution. For general supervised learning tasks, our goal is to find an optimal model function that minimizes a given loss function on the training dataset. In contrast, the ultimate optimization objective of reinforcement learning tasks is to maximize the value of the agent's policy during interaction with the dynamic environment:

$$\pi = \arg\min_{\pi} \mathrm{E}_{(S,A)\sim\pi}[R(S,A)] \tag{1}$$

where π is the strategy to be learned, S is the current state of the environment, A is the action taken by the intelligent agent, and $R(S,A)$ is the reward obtained from action A in state S.

II. A. 2) Markov Decision Process

The concept of Markov decision processes [36] originated from Markov chains. Markov chains are used to describe the state transition rules in stochastic processes. The conditional probability distribution of the current state transitioning to the future state is only related to the current state and is not affected by past states:

$$P(S_{t+1} \mid S_t) = P(S_{t+1} \mid S_1,\ldots,S_t) \tag{2}$$

In the S_t-hour random process, the random phenomenon at a certain time t, $(S_1,\ldots,S_t)$ is known historical information, $P(S_t+1 \mid S_t)$ is the probability that the value at time t is S_t and the state at the next time is S_{t+1}.

The core of reinforcement learning lies in learning the optimal strategy through interaction with the environment, and the Markov decision process (MDP) provides the theoretical foundation for this learning process. The first step in solving a practical problem using reinforcement learning is to convert the practical problem into an MDP, which requires clarifying the following key components: $\langle \mathrm{S},\mathrm{A},P,r,\gamma \rangle$, where:

S is the set of system environment states;

A is the set of actions that the intelligent agent can take in the environment;

γ is the discount factor;

r is the reward function $r(s,a)$, representing the reward that the agent can obtain when executing action a in state s; P is the state transition function $P(s' \mid s,a)$, representing the probability of the agent reaching state s' after executing action a in state s.

In addition to the two main components—the environment and the agent—a Markov decision process requires three additional elements: the agent's policy, the state value function based on the policy, and the action value function obtained by executing the policy.

The agent's policy π is the rule or model for selecting actions in each state. The policy $\pi(a \mid s) = P(A_t = a \mid S_t = s)$ represents the probability of taking action a given the input state s.

The state value function $V^{\pi}(s)$ represents the expected cumulative reward that the agent would receive in state s by following a certain policy π in the Markov decision process. That is:

$$V^{\pi}(s) = \mathrm{E}_{\pi}[G_t \mid S_t = s] \tag{3}$$

The action value function $Q^{\pi}(s,a)$ evaluates the expected cumulative reward of taking action a in state s according to policy π. That is:

$$Q^{\pi}(s,a) = \mathrm{E}_{\pi}[G_t \mid S_t = s, A_t = a] \tag{4}$$

The relationship between the state value function and the action value function is as follows:

$$V^{\pi}(s) = \sum_{a\in A} \pi(a \mid s) Q^{\pi}(s,a) \tag{5}$$

When using strategy π, the value of taking action a in state s is:

$$Q^{\pi}(s,a) = r(s,a) + \gamma \sum_{s'\in S} P(s' \mid s,a) V^{\pi}(s') \tag{6}$$

II. A. 3) Strategy optimization methods based on deep reinforcement learning

Deep Reinforcement Learning combines the neural network technology of deep learning with the policy optimization methods of reinforcement learning. It uses neural networks to model the behavior strategies of agents, with the core idea being to learn optimal behavior strategies through extensive data interaction. Deep Reinforcement Learning applies deep learning to reinforcement learning, particularly in problems with large state spaces or action spaces. By using deep neural networks, DRL can handle high-dimensional inputs such as images or text to approximate optimal policies or value functions. The following are some key features of DRL: Function approximation: Using neural networks to approximate value functions (such as Q-values) or policy functions, thereby reducing the need for manual feature engineering. Experience replay: Breaking the correlation between data by storing and reusing past experiences to enhance learning efficiency. Target network: Introduces two networks, one for generating target values and another for updates, to improve stability and convergence. Currently, deep reinforcement learning algorithms are primarily divided into two categories: value-based deep reinforcement learning algorithms and policy gradient-based deep reinforcement learning algorithms.

II. B. ***Building Automatic Layout Model Based on Reinforcement Learning***

II. B. 1) Overall Framework Design

A multi-scenario building automatic layout method based on deep reinforcement learning. After initially determining the building height and number of buildings based on the floor area ratio of the residential area, the method can input the buildings into the model as an initial layout by randomly or empirically placing them. Each building is treated as an agent. At step t, the observation of the environment is s_t, and the current reward value r_i is calculated based on the set constraints. The current environment, action strategy, reward value, and new environment are packaged into a tuple (s_i, a_i, r_i, s_{i+1}) and stored in the experience pool. This process continuously accumulates experience from different layouts, which is used to train the deep reinforcement learning network for self-learning and self-optimization. Gradually, the model identifies building layouts that satisfy the constraints and achieve optimal building sunlight exposure duration, thereby achieving a building layout automatic design model that meets multiple constraints and is applicable to various scenarios.

II. B. 2) State Space and Action Space Design

The state space s of intelligent agent observation is defined as:

$$s = [v_x, v_y, x - x_i, y - y_i] \tag{7}$$

In the equation: v_x, v_y represent the current agent's velocity along the x and y axes; x_i, y_i represent the current agent's position information; $x = \{x_1, \cdots, x_N\}$ and $y = \{y_1, \cdots, y_N\}$ represent the position information of all buildings in the scene.

The movement of the intelligent agent is a deterministic behavior, and the action space a of the intelligent agent is defined as:

$$a = [Noop, a_x^+, a_x^-, a_y^+, a_y^-] \tag{8}$$

In the equation: Noop is unused, representing no operation; a_x^+, a_x^- and a_y^+, a_y^- represent the acceleration of the current intelligent agent in the positive and negative directions of the x and y axes, respectively.

The default period of the environment is $T = 0.1s$. Thus, the updated velocities $v_x^{'}, v_y^{'}$ of the intelligent agent can be calculated as:

$$v^{'}_x = v_x + (a_x^+ - a_x^-) * T \tag{9}$$

$$v^{'}_y = v_y + (a_y^+ - a_y^-) * T \tag{10}$$

The updated position coordinates of the intelligent body are:

$$x' = x + v'_x * T \tag{11}$$

$$y' = y + v'_y * T \tag{12}$$

When an intelligent agent moves and the coordinates of a building unit change, the state space observed by the intelligent agent also changes accordingly.

II. B. 3) Constraint Conditions and Reward Function Design

(1) Constraints imposed by the land use boundary line

The land use boundary line is the boundary line of the land use rights area. The constraints imposed by the land use boundary line are expressed as follows:

$$C_{landuse} = \begin{cases} 0, & N_1 = 0 \\ -10000, & N_1 \neq 0 \end{cases} \tag{13}$$

(2) Fire separation distance constraints

To determine whether the building layout complies with fire separation distance regulations, the constraints are expressed as follows:

$$C_{fire} = \begin{cases} 0, & A_{fire} = 0 \\ -A_{fire}, & A_{fire} \neq 0 \end{cases} \tag{14}$$

The fire separation distance constraint value C_{fire} defined by the above design can be used to constrain collision issues between buildings. If a collision occurs between buildings, a penalty value of $-A_{fire}$ will be applied.

(3) Constraints and optimization of sunlight exposure duration

The two vertices and the center point of the southbound boundary line of the building unit are used as sunlight exposure test points. The calculation method for the duration of sunlight exposure test point obstruction is as follows:

(i) Between 8:00 and 11:00, the shadow length L and shadow azimuth A' at the corresponding time. The lower left vertex $P_{ld} = (x_{ld}, y_{ld})$ of building n is obtained by the following formula: the vertex $S_{ld} = (x'_{ld}, y'_{ld})$ corresponding to the shadow of the lower left vertex, that is:

$$x'_{ld} = x_{ld} + \cos(3\pi/2 - A'_s * \pi/180) * L \tag{15}$$

$$y'_{ld} = y_{ld} + \sin(3\pi/2 - A'_s * \pi/180) * L \tag{16}$$

(ii) At time 12:00, calculate the vertices S_{lu}, S_{ru} corresponding to the shadows of the upper left vertex P_{lu} and upper right vertex P_{ru} of building n according to equations (15) and (16). At this time, point $P_{lu}, S_{lu}, S_{ru}, P_{ru}$ is the vertex sequentially forming the polygonal area, which is the shadow area S_j^n of building n at time j, which is 12:00.

(iii) From 13:00 to 16:00 at the moment, calculate the lower right vertex P_{nd}, the upper right vertex P_{nd}, and the upper left vertex P_{lu} of building n according to equations (15) and (16) to determine the three vertices corresponding to the three vertices S_{rd}, S_{ru}, S_{lu} of the shadow. At this point, the polygonal area formed sequentially with point $P_{lu}, S_{lu}, S_{ru}, S_{rd}, P_{ru}$ as the vertex is the shadow area S_j^n of building n at time j, which is 13:00 to 16:00.

(4) Additional constraints for multiple scenarios

Under the automatic layout optimization of buildings in complex scenarios, the sum of the building areas overlapping with the forbidden range is determined to be A_{forbid}, and its constraints are expressed as follows:

$$C_{addition} = \begin{cases} 0, & A_{forbid} = 0 \\ -A_{forbid}, & A_{forbid} \neq 0 \end{cases} \tag{17}$$

(5) Reward function design

Add up all the results, and when the total constraint condition value C_{all} is 0, a reasonable high-rise residential building layout plan can be output:

$$C_{all} = C_{sunshine} + C_{fire} + C_{landuse} + C_{addition} \tag{18}$$

In the design of the model's reward function r_t, the total duration of shading of the sunlight test points for buildings awaiting demolition within the plot L_{shadow} is included, i.e.:

$$r_t = C_{sunshine} + C_{fire} + C_{landuse} + C_{addition} - L_{shadow} \tag{19}$$

II. B. 4) Deep Deterministic Policy Gradient Algorithm

The continuous action values, state space values, and reward values of a building moving in the environment are input into the deep deterministic policy gradient (DDPG) algorithm [37]. The specific process of the algorithm is as follows:

1) The parameters of the actor and critic in the main network are randomly initialized as θ^{μ}, θ^{Q}, the Actor obtains the mapping from state to action $\mu(s \mid \theta^{\mu})$, and the Critic obtains the Q-value $Q(s, a \mid \theta^{Q})$

2) Assign the parameters of the main network to the target network for parameter initialization, $\theta^{\mu} \to \theta^{\mu'}, \theta^{Q} \to \theta^{Q'}$, and initialize the experience pool at the same time.

3) Initialize the layout state s_1. The Actor of the main network selects action a_t for exploration based on the current policy μ plus Gaussian perturbation N_t:

$$a_t = \mu(s \mid \theta^{\mu}) + N_t \tag{20}$$

4) After executing action a_t, the corresponding reward r_t and the next state s_{t+1} are obtained, and the tuple (s_t, a_t, r_t, s_{t+1}) as a sampling value and store it in the experience pool as the dataset for training the main network.

5) Randomly sample N sample data points (s_t, a_t, r_t, s_{t+1}) from the experience pool, which serve as a mini-batch of training data for the main network. The current Q value is estimated using the Bellman equation:

$$y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1} \mid \theta^{\mu'}) \mid \theta^{Q'}) \tag{21}$$

In the formula: γ represents the discount rate, which indicates the degree of influence of estimated future rewards on current decisions. If $\gamma = 0$, then no future value is considered; if $\gamma = 1$, then future value does not decay over time.

6) Calculate the loss function L by taking the difference between the estimated value of Q and the actual value, and use gradient descent to update the Critic network:

$$L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i \mid \theta^Q))^2 \tag{22}$$

7) Update the Actor network using the strategy gradient method:

$$\nabla_{\theta^\mu} J(\theta) \approx \frac{1}{N}\sum_i \nabla_a Q(s, a \mid \theta^0)\big|_{s=s_i, a=\mu(s_i)} \times \nabla_{\theta^\mu} \mu(s \mid \theta^\mu)\big|_{s_i} \tag{23}$$

8) After several iterations, assign the parameters of the main network to the target network and update the target network parameters:

$$\begin{cases} \theta^{Q'} \leftarrow \kappa\theta^Q + (1-\kappa)\theta^{Q'} \\ \theta^{\mu'} \leftarrow \kappa\theta^\mu + (1-\kappa)\theta^{\mu'} \end{cases} \tag{24}$$

II. C.Analysis of the automatic generation of building layouts based on the DDPG algorithm

This paper sets up an experimental scenario for a site requiring solar optimization, which includes multi-story buildings to be arranged, existing buildings, and a simulated surrounding environment outside the site. After designing the model, experiments are conducted. In the state space, buildings are represented by two coordinate points, with one point set at the center of the building's north-south axis. The experiment involves six buildings to be arranged. Initially, the initial arrangement coordinates of the six buildings are input into the model, along with their initial layout states. The initial arrangement of the multi-story buildings on the site is shown in Figure 1. After the six buildings are arranged to satisfy the solar access constraints, the building coordinates and corresponding visualization experimental results are obtained.

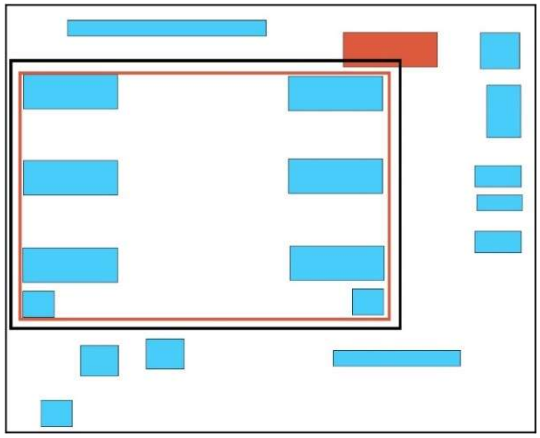


Figure 1: The initial layout of multi-storey buildings in the site

The optimized building coordinates for multi-story buildings are shown in Table 1. The visualization results of the multi-story building solar access optimization scheme are shown in Figure 2, where (a) to (e) represent the visualization results of optimization schemes 1 to 5, respectively. In all experimental results, after solar radiation calculations, each multi-story building to be arranged within the site meets the minimum solar radiation constraints. In the optimized schemes obtained based on solar radiation constraints, the selection of the scheme is made by professionals in the architectural field.

Table 1: Multi-layer buildings are optimized after the sun is optimized

Optimization plan	X-axis	Y-axis
Multi-layer building sunshine optimization scheme 1 coordinate	0.4482	0.3353
	-0.0906	0.3069
	-0.5492	0.213
	0.0992	-0.1654
	-0.5044	-0.3125
	-0.0089	-0.6092
Multi-layer building sunshine optimization scheme 2 coordinate	0.5827	0.4648
	-0.0065	0.4625
	-0.5554	0.2752
	0.1857	-0.1589
	-0.4873	-0.1316
	-0.3391	-0.497
Multi-layer building sunshine optimization scheme 3 coordinate	0.5233	0.3127
	-0.059	0.2258
	-0.5809	0.4834
	0.5694	-0.2259
	-0.5522	-0.3241
	0.0359	-0.5935
Multi-layer building sunshine optimization scheme 4 coordinate	0.4866	0.2217
	-0.0506	0.1364
	-0.5708	0.3402
	0.5619	-0.1911
	-0.5535	-0.247
	0.1513	-0.6031
Multi-layer building sunshine optimization scheme 5 coordinate	0.5304	0.4519
	0.0189	0.4344
	-0.4742	0.4455
	0.2059	-0.0904
	-0.5944	-0.3929
	-0.1006	-0.6008

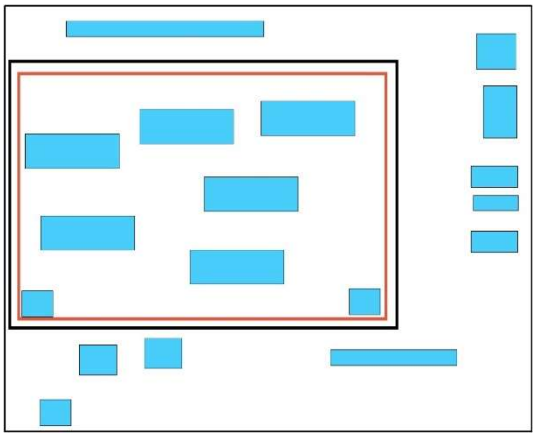
II. C. 1) Training Data Set Creation

Three different types of training datasets were created for comparison: Mode A dataset: residential area contour images (color block diagram) and master plan of the strong layout scheme (figure-ground relationship diagram); Mode B dataset: residential area contour images (contour lines) and master plan diagrams of the forced layout scheme (figure-ground relationship diagram); Mode C dataset: residential area contour images (color block diagrams) and master plan diagrams of the forced layout scheme (satellite images). Each dataset sample in these modes represents the corresponding relationship between the residential area contours and the master plan diagrams.

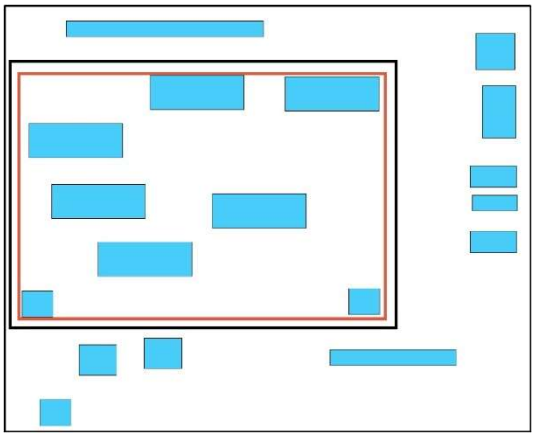
Different mode datasets were used to train the network model. After training, images were selected from the test set for evaluation. The generated images were comprehensively evaluated based on clarity, realism, and feasibility, and the mode with better generated image results was selected for subsequent training dataset creation. For evaluating the realism of generated images, the Image Structure Similarity Index (SSIM) algorithm was applied to calculate the similarity between the real image and the image generated by the network model. This study categorizes major cities nationwide into three types. Residential areas in major cities located in the mid-latitude region (latitude range 30°N to 38°N) were selected as training dataset samples, totaling 11 cities. The selected residential area samples in the dataset have an area range of 15,300 to 154,100 m², with buildings predominantly arranged in a grid-like layout, and some combining grid-like and perimeter-style layouts.

The original data for the training dataset was obtained from city shapefile format files on the CSDN website. The open-source geographic information tool QGIS was used to read the city shapefile format files. Based on the correspondence between building location attributes and floor information, residential areas were classified, and samples of each residential area category were screened according to floor area ratio. Floor area ratio is an important indicator reflecting the intensity of three-dimensional land development. Low-rise residential areas with a FAR > 0.5 were defined as low-rise high-density residential areas, and multi-story residential areas with a FAR between 1.2 and 1.6 were defined as multi-story high-density residential areas. The study selected high-density

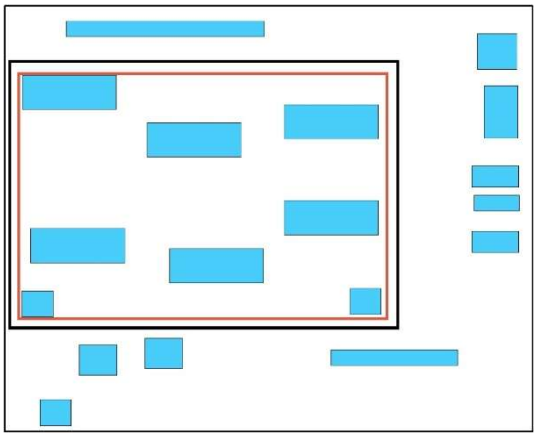
residential area samples with a FAR > 0.5 for low-rise residential areas, a FAR > 1.5 for multi-story residential areas, and a FAR > 2.5 for high-rise residential areas to create the training dataset.



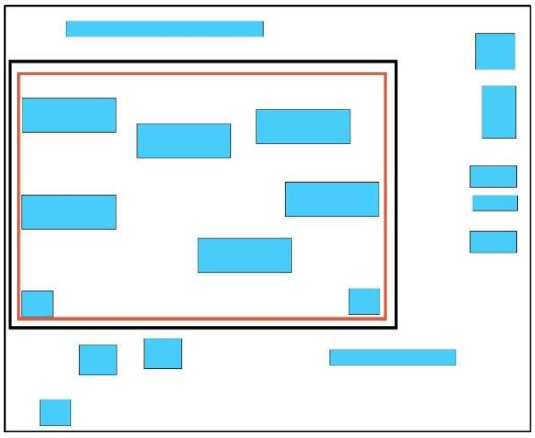
(a) The visual results of optimization plan 1



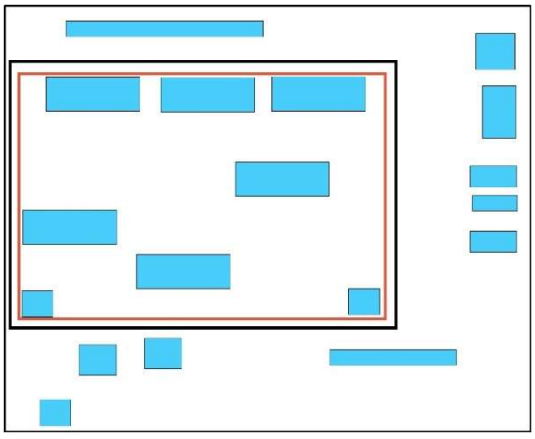
(b) The visual results of optimization plan 2



(c) The visual results of optimization plan 3



(d) The visual results of optimization plan 4



(e) The visual results of optimization plan 5

Figure 2: Visualization of the optimization scheme of multi-layer buildings

III. Analysis of the effectiveness of urban building functional zoning optimization based on CGAN

III. A. Design of residential area forced drainage schemes based on CGAN

The design process for generating a forced drainage plan for residential areas based on CGAN is shown in Figure 3. Taking an actual residential area plot as an example, its contour image is used as the model input to generate a forced drainage design plan. In the parametric modeling tool, the geometric model of the residential area buildings is constructed based on the correspondence between the pixel gray values and the number of layers in the generated forced drainage design plan. The geometric model of the residential area buildings is then subjected to a solar radiation simulation analysis to validate and evaluate the generated plan.

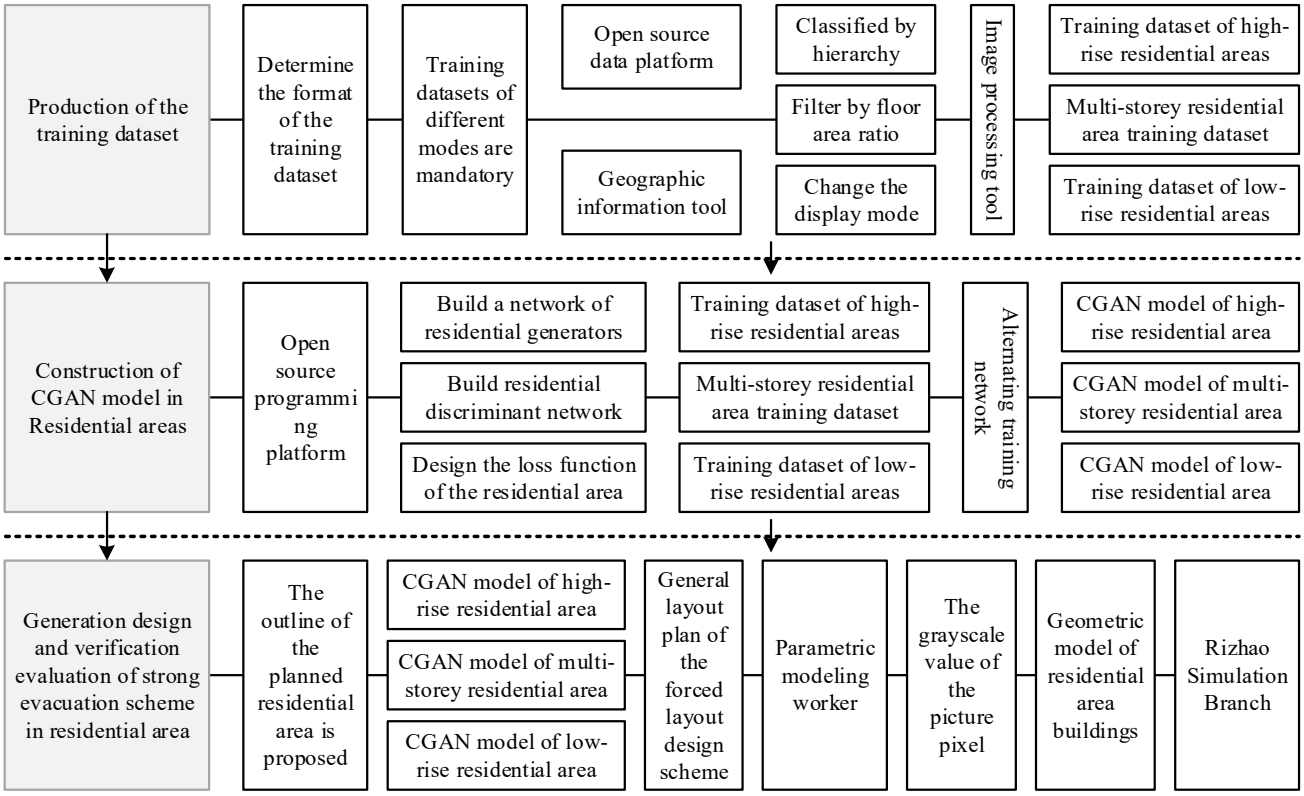


Figure 3: The residential area is a strong line scheme to generate the design process

III. A. 1) Construction of the CGAN model for residential areas

The residential area CGAN model consists of a residential area generator network and a residential area discriminator network, which engage in a competitive game against each other. The input of the residential area generator network is the residential area contour image, and the output is the master plan of the residential area layout scheme. The loss function for the residential area generator and discriminator networks is based on the CGAN loss function and incorporates an L1 loss function to enhance the accuracy and realism of the generated master plan for the residential area layout.

The CGAN loss function is:

$$L_{CGAN}(G,D) = E_{x,y}[\log D(x,y)] + E_{x,z}[\log(1 - D(x,G(x,z))] \tag{25}$$

The L1 loss function can improve the accuracy and realism of the master plan for residential area drainage schemes, and can be expressed as:

$$L_{1,1}(G) = E_{x,y,z}[\| y - G(x,z) \|_1] \tag{26}$$

The final residential area CGAN loss function is:

$$G^* = \arg\min_G \max_D L_{CGAN}(G,D) + \lambda L_{LJ}(G) \tag{27}$$

In the equation: G is the residential area generator network; D is the residential area discriminator network; x is the residential area contour image; y is the actual residential area strong-exclusion plan master plan;

$G(x,z)$ is the generated image from the residential area generator network; z is the input random noise, replaced by dropout; λ is the set weight value.

The study was conducted using the open-source programming platform Anaconda, the deep learning framework TensorFlow, and the interactive editing tool Jupyter Notebook to construct CGAN models for high-rise, multi-story, and low-rise residential areas. The model's input is the residential area contour image, and the output is the master plan of the residential area's forced relocation scheme. Model training parameters are divided into optimizer hyperparameters and model hyperparameters, with model hyperparameters using the default parameter settings from the pix2pix algorithm. Considering both image quality and training time costs, the number of iterations in the optimizer hyperparameters is set to 500, and the initial learning rate is set to 0.0001.

III. B. Analysis of residential group layout results based on deep learning

III. B. 1) Experimental Design

Taking a six-story residential building as an example, divide it into three datasets with 200, 400, and 800 units, respectively. Test after each round of training, compare the results generated in each round, record the loss of the generator and discriminator in each round, and plot it into a line chart.

III. B. 2) Generation Process and Results Analysis

The generation process mainly consists of the following steps:

Step 1: Create a dataset. Divide the six-story residential buildings into datasets containing 200, 400, and 800 data points, respectively. Select condition images that do not appear in the training set as the test set to test the performance of the generated model.

Step 2: Input data for training. Match the target images and conditional images of the layout scheme into pairs and feed them into the pix2pix network. Each dataset undergoes 20 rounds of training.

Step 3: Test the generated model, printing the generated results every two rounds.

The loss of the generator and discriminator, as well as the accuracy of the discriminator, are shown in Figure 4. As can be seen from the figure, overall, the loss of the generator and the loss of the discriminator both decrease as the number of training rounds increases. The loss of the discriminator stabilizes after a certain number of rounds, while the loss of the generator decreases with fluctuations, so the generation results fluctuate to some extent as the number of rounds increases. The accuracy rate of the discriminator fluctuates significantly, with larger data volumes leading to greater fluctuations, and gradually improves over time. From the generated results, the model with a data volume of 200 achieves relatively good results after 11 training rounds, the dataset with a data volume of 400 achieves the best results after 10 training rounds, and the dataset with a data volume of 800 achieves the best results after 5 training rounds. It can be seen that the more data there is, the fewer rounds are needed to achieve relatively good results. From the best results generated by each dataset, it can be seen that the more data there is, the better the best results generated. Therefore, when the training results are poor, the training results can be improved by increasing the amount of data. When the amount of data is limited, the model's performance can be improved by increasing the number of training rounds. However, when the amount of data is small, even increasing the number of training rounds cannot achieve the best results that can be achieved with a large amount of data.

III. C. Analysis of the optimization design effects of functional zoning in building layout

All images in this dataset are 256×256 pixels in size. This paper simplifies the classification of urban spatial buildings, dividing them into two main categories: residential buildings and recreational buildings, which include three common functional zones: “green areas, supporting service areas, and living areas.”

By calculating according to the specified indicators, the floor area of buildings under the corresponding urban scale can be determined. During calculation, only the floor area indicators required for buildings in different-tier cities are included.

By counting the number of RGB value pixels representing a specific function in the image and the number of RGB value pixels within the land use area, the proportion of pixels for each functional zone can be obtained.

(1) Residential building floor area ratio

When calculating the floor area of residential buildings in the dataset, the following methods are used: residential buildings in the central area are calculated based on a 21-story building, while other residential buildings are calculated based on a 6-story building. The comparison results of residential building floor areas between the training set and the test set are shown in Figure 5. The results show that the average values for the training set and test set are 263,604 m² and 250,126 m², respectively, with the training set slightly higher than the test set. This is partly because the land area of the training set is slightly larger than that of the test set. The reason is that the training data is compressed before being input into the computer for learning, which may cause inaccuracies in the RGB values of some building pixels, leading to information loss during model learning. This loss results in slightly fewer building pixels in the final model-generated images. However, in terms of the relationship between land area

and residential building floor area, the distributions of the training set and test set are basically consistent, indicating that the overall learning performance of the model is still quite good. Ultimately, 25.67% of the residential building floor area in the training set meets the recommended indicators, while 74.33% exceeds them; in the test set, 38.74% of the residential building floor area meets the recommended indicators, while 61.26% exceeds them, which generally meets the requirements.

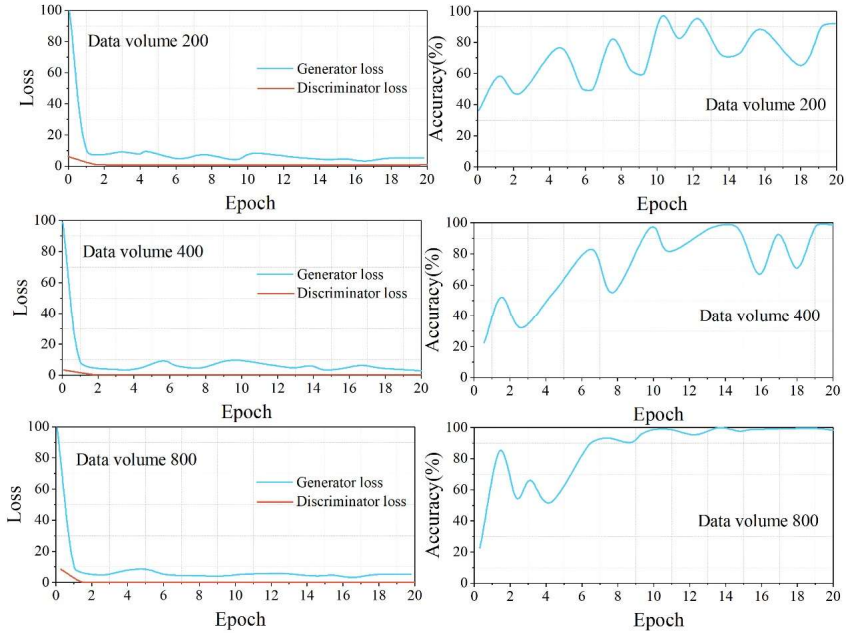


Figure 4: The generator, the discriminator loss and the accuracy of the discriminator

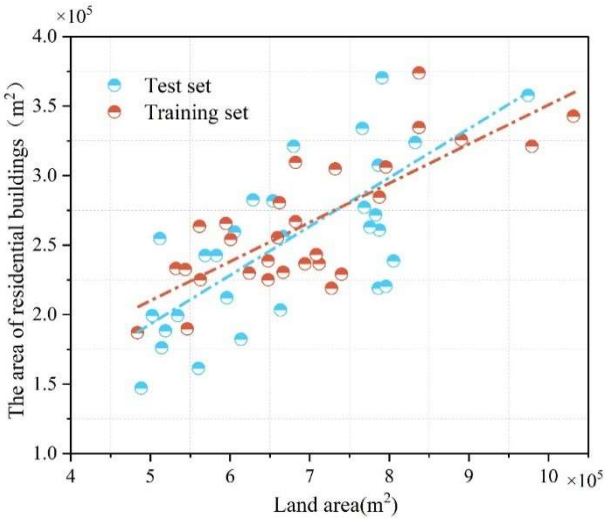


Figure 5: The training set is compared with the test set teaching building area

(2) Leisure building area ratio

When calculating the area of leisure buildings (parks or amusement parks) in the dataset, all parks were calculated based on an area of 5,000 square meters. The comparison results of the leisure building areas in the training set and test set are shown in Figure 6. The average values for the training set and test set are 697,358 square meters and 662,091 square meters, respectively, with a fluctuation of no more than 10%, which is within a reasonable range. In terms of the relationship between land area and recreational building area, the distributions of the training set and test set are relatively consistent. However, under the same land area, the training set shows a significant number of cases where the recreational building area is lower than that of the test set, indicating that the

model has some errors when generating the recreational building portion. Compared with the recommended indicators, 62.15% of the residential building area in the training set meets the recommended indicators, while 37.85% is below the recommended indicators; 45.22% of the residential building area in the test set meets the recommended indicators, while 54.78% is below the recommended indicators, with an overall tendency toward smaller values. The reasons are as follows: 1. The training set already had cases with relatively small recreational area sizes during annotation. 2. Compared to other buildings, recreational buildings are more sparsely distributed and irregularly arranged, and some indoor areas are easily confused with residential buildings. Therefore, errors are more likely to occur in the output data, thereby affecting the completeness and clarity of recreational buildings.

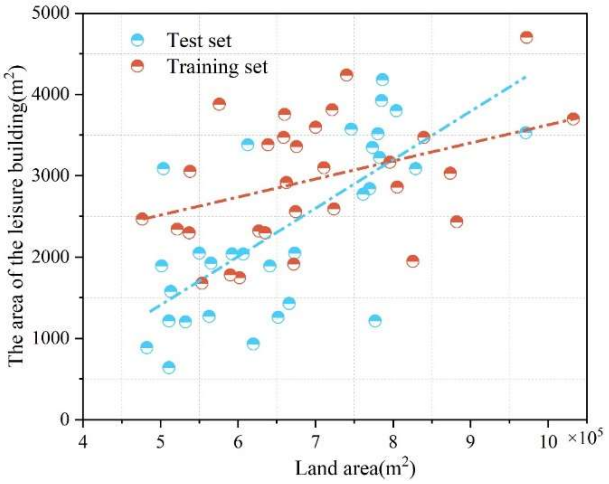


Figure 6: The training set is compared with the test set leisure building area

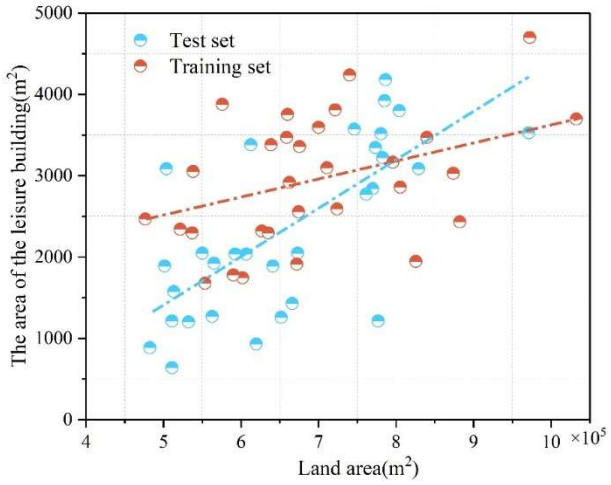
(3) Proportion of land area for each functional zone

By counting the number of RGB value pixels representing a specific function in the image and the number of RGB value pixels within the land area, we can obtain the proportion of pixels for each functional zone, which is essentially the proportion of land area. Based on this logic, we used Python code to calculate the proportion of land area for each functional zone in the dataset and found that the average area proportions of each functional zone in the test set and training set are largely consistent, enabling further analysis.

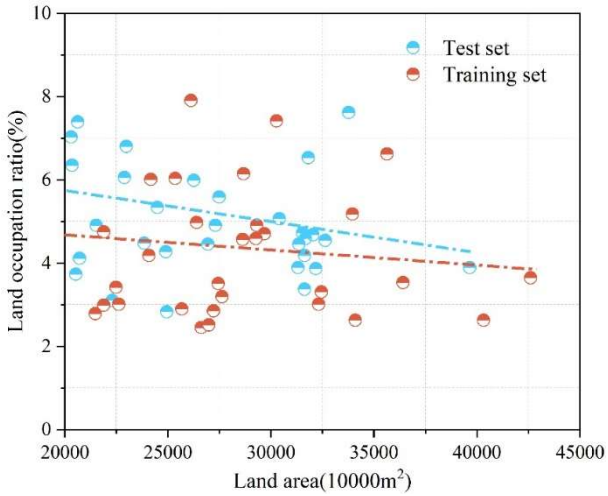
The comparison of land area proportions across the three functional zones between the training set and test set is shown in Figure 7, where (a) to (c) represent the land areas of the green zone, transportation zone, and residential zone, respectively. The results indicate that the distribution trends of the training set and test set are consistent across the three zones. The proportion of land area in the green space functional zone decreases as the total land area of the city increases. This is because as the city's population grows, the demand for land to meet residential needs increases. Under the premise of basically meeting the city's green space requirements, this further drives the conversion of other available land in the city into residential, industrial, and other commercially valuable functional zones to meet people's survival needs. Meanwhile, the proportion of land area allocated to supporting service functional zones and residential zones generally shows a subtle upward trend as the total land area of the city increases, but overall remains relatively stable with minimal changes in proportion. This aligns with the principle that as the scale of urban development grows and the population increases, the area of living space provided increases proportionally.

(4) Building density and floor area ratio

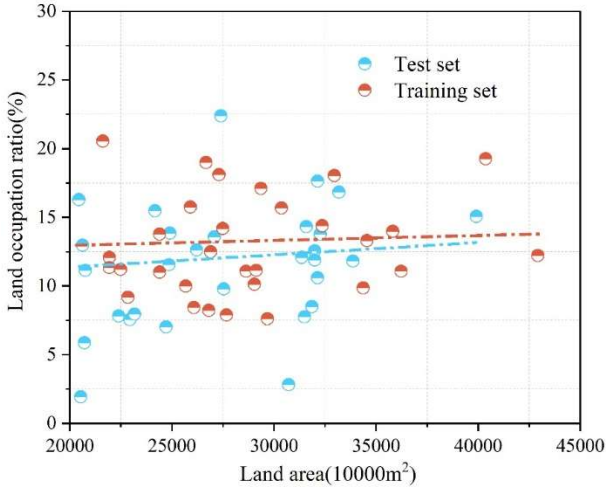
Research indicates that urban building density typically remains below 25%, with most cities averaging around 20%. Urban spatial planning and construction should determine land use based on the total floor area and corresponding floor area ratio of various functional zones, with a recommended floor area ratio of 0.5, without specifying building density. Statistical analysis of the floor area ratio in the dataset reveals that the average floor area ratio in the training set is 0.5329, with a maximum value of 0.6925 and a minimum value of 0.4608; the average floor area ratio in the test set is 0.5217, with a maximum value of 0.6458 and a minimum value of 0.3746. Compared to the recommended indicator of 0.5, the floor area ratio of the test set data generally aligns with the recommended indicator.



(a)The area of land in the green area



(b)The area of land in the traffic area



(c)The area of land in the living area

Figure 7: The training set is compared to the area of the test set

IV. Conclusion

This paper constructs an automatic urban building layout model based on reinforcement learning and uses the CGAN model to analyze and verify the effectiveness of the generated urban residential area layout schemes.

(1) The results of the sunlight calculation show that each multi-story building to be laid out in the plot meets the minimum sunlight constraint, indicating that the automatic building layout generated based on the DDPG algorithm meets the requirements of this paper.

(2) When the dataset size is 800, the model achieves optimal performance after only 5 training rounds, reducing the number of training rounds by over 50% compared to a dataset size of 200. This demonstrates that as the dataset size increases, the number of training rounds decreases, leading to better experimental results.

(3) An evaluation of the generated scheme's indicators across six aspects—land area, the ratio of residential and recreational building areas, the proportion of land area for the three functional zones (“green areas, supporting service areas, and residential areas”), and floor area ratio—revealed that the final experimental results exhibit reasonable functional relationships, meet training requirements, and generally align with regulatory recommendations, confirming that the experiment achieved satisfactory outcomes.

Funding

Anhui Provincial Social Science Fund (8501008620).

References

[1] Peng, J., & Peng, F. L. (2018). A GIS-based evaluation method of underground space resources for urban spatial planning: Part 1 methodology. Tunnelling and Underground Space Technology, 74, 82-95.
[2] Hersperger, A. M., Oliveira, E., Pagliarin, S., Palka, G., Verburg, P., Bolliger, J., & Grădinaru, S. (2018). Urban land-use change: The role of strategic spatial planning. Global Environmental Change, 51, 32-42.
[3] Hersperger, A. M., Grădinaru, S., Oliveira, E., Pagliarin, S., & Palka, G. (2019). Understanding strategic spatial planning to effectively guide development of urban regions. Cities, 94, 96-105.
[4] Zawidzki, M. (2016). Discrete optimization in architecture: Architectural & urban layout. Singapore: Springer.
[5] Fan, Q., Mei, X., Zhang, C., & Yang, X. (2022). Research on Gridding of Urban Spatial Form Based on Fractal Theory. ISPRS International Journal of Geo-Information, 11(12), 622.
[6] Bao, M., & Zou, D. (2024). The Impact of Architectural Layout on Urban Heat Island Effect: A Thermodynamic Perspective. International Journal of Heat & Technology, 42(2).
[7] Vermeulen, T., Knopf-Lenoir, C., Villon, P., & Beckers, B. (2015). Urban layout optimization framework to maximize direct solar irradiation. Computers, Environment and Urban Systems, 51, 1-12.
[8] Dong, X. (2023). The Impact of Urban Planning Management on Urban Architectural Design. Landscape and Urban Horticulture, 5(1), 42-46.
[9] Zakirova, Y. A., Khusnutdinova, S. R., & Faizrakhmanova, G. R. (2020, February). Transformation of the city environment: functional-planning and territorial aspects. In IOP Conference Series: Materials Science and Engineering (Vol. 753, No. 4, p. 042069). IOP Publishing.
[10] Burlacu, S., Gavrilă, A., Popescu, I. M., Gombos, S. P., & Vasilache, P. C. (2020). Theories and models of functional zoning in urban space. Revista de Management Comparat International, 21(1), 44-53.
[11] Dubrova, S. V., Podlipskiy, I. I., Kurilenko, V. V., & Siabato, W. (2015). Functional city zoning. Environmental assessment of eco-geological substance migration flows. Environmental Pollution, 197, 165-172.
[12] Rodrigues, C., Veloso, M., Alves, A., & Bento, C. L. (2025). Socioeconomic and functional zoning characterization in a city: a clustering approach. Cities, 163, 106023.
[13] Cortinovis, C., Haase, D., Zanon, B., & Geneletti, D. (2019). Is urban spatial development on the right track? Comparing strategies and trends in the European Union. Landscape and urban planning, 181, 22-37.
[14] Williams, K. (2017). Spatial planning, urban form and sustainable transport: An introduction. In Spatial planning, urban form and sustainable transport (pp. 15-28). Routledge.
[15] Bartz-Beielstein, T., Branke, J., Mehnen, J., & Mersmann, O. (2014). Evolutionary algorithms. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 4(3), 178-195.
[16] Wang, D., Tan, D., & Liu, L. (2018). Particle swarm optimization algorithm: an overview. Soft computing, 22(2), 387-408.
[17] Liu, M., He, Y., Wang, J., Lee, H. P., & Liang, Y. (2015). Hybrid intelligent algorithm and its application in geological hazard risk assessment. Neurocomputing, 149, 847-853.
[18] Ko, J., Ennemoser, B., Yoo, W., Yan, W., & Clayton, M. J. (2023). Architectural spatial layout planning using artificial intelligence. Automation in Construction, 154, 105019.
[19] AlOmani, A., & El-Rayes, K. (2020). Automated generation of optimal thematic architectural layouts using image processing. Automation in construction, 117, 103255.
[20] Bibri, S. E. (2021). Data-driven smart sustainable cities of the future: Urban computing and intelligence for strategic, short-term, and joined-up planning. Computational Urban Science, 1(1), 8.
[21] Zhao, H., Jiang, X., Gu, B., & Wang, K. (2022). Evaluation and Functional Zoning of the Ecological Environment in Urban Space—A Case Study of Taizhou, China. Sustainability, 14(11), 6619.
[22] Zhao, Y., Leng, H., Sun, P., & Yuan, Q. (2018). A spatial zoning model of municipal administrative areas based on major function-oriented zones. Sustainability, 10(9), 2976.
[23] Wang, S., Qu, Y., Zhao, W., Guan, M., & Ping, Z. (2022). Evolution and optimization of territorial-space structure based on regional function orientation. Land, 11(4), 505.
[24] Xu, X., Na, R., Shen, Z., & Deng, X. (2023). Impact of major function-oriented zone planning on spatial and temporal evolution of “three zone space” in China. Sustainability, 15(10), 8312.
[25] Li, Y., Chen, Y., Zhao, M., & Zhai, X. (2018). Optimization of planning layout of urban building based on improved logit and PSO algorithms. Complexity, 2018(1), 9452813.
[26] Dino, I. G., & Üçoluk, G. (2017). Multiobjective design optimization of building space layout, energy, and daylighting performance. Journal of Computing in Civil Engineering, 31(5), 04017025.
[27] Fan, Q., Mei, X., Zhang, C., & Wang, H. (2022). Urban spatial form analysis based on the architectural layout--Taking Zhengzhou City as an example. Plos one, 17(12), e0277169.
[28] Gurcan Bahadir, C. G., & Tong, T. (2025). Computational approaches to space planning: A systematic review of enhancing architectural layouts. International Journal of Architectural Computing, 14780771241310215.
[29] Biao, L., Cunyan, J., Lu, W., Weihua, C., & Jing, L. (2019). A parametric study of the effect of building layout on wind flow over an urban area. Building and Environment, 160, 106160.
[30] Tong, W. U., Wenze, Y. U. E., Haoxuan, X. I. A., & Jinhui, X. I. O. N. G. (2022). Optimization of major function zoning strategy from the perspective of territorial spatial planning. Economic geography, 42(2), 11-17.
[31] Zhang, W., & Li, B. (2021). Research on an analytical framework for urban spatial structural and functional optimisation: a case study of Beijing city, China. Land, 10(1), 86.
[32] Dong, W., Wu, N., Xiang, G., Wu, X., & Li, J. (2025). Spatial Data Analysis and Urban Functional Zoning Optimization in Smart City Development. International Journal of High Speed Electronics and Systems, 2540634.

[33] Zhao, J., & Zhao, J. (2015). Research on Spatial Coupling of the Urban Planning Based on Master and Main Functional Area. Applied Mechanics and Materials, 716, 280-283.
[34] Chen, Y., & Xu, F. (2022). The optimization of ecological service function and planning control of territorial space planning for ecological protection and restoration. Sustainable Computing: Informatics and Systems, 35, 100748.
[35] Chen, Y., Yang, J., Yang, R., Xiao, X., & Xia, J. C. (2022). Contribution of urban functional zones to the spatial distribution of urban thermal environment. Building and Environment, 216, 109000.
[36] Yang Jiang. (2025). Design of dynamic career path recommendation system based on Markov decision process. Journal of Computational Methods in Sciences and Engineering,25(3),2427-2441.
[37] Fei Zhou,Lihong Zhao,Xiaomei Ding & Shuqin Wang. (2025). Enhanced DDPG algorithm for latency and energy-efficient task scheduling in MEC systems. Discover Internet of Things,5(1),40-40.