Eng Int Syst (2026) 5: 627–639
© 2026 CRL Publishing Ltd

Engineering
Intelligent Systems

Metaverse-oriented Multi-task Learning Algorithms in Animation Scene Design

Zhuoyi Song, Xinyu Guo and Yu Gu*

Jiamusi University, Jiamusi 154007, Heilongjiang, China

In response to the problems of poor efficiency in animation scene design and the inadequate immersive experience of users , this study focuses on the metaverse and takes the animation *Spirited Away* as the object, applying multi-task learning algorithms to examine animation scene design. Firstly, a multi-task learning framework based on generative adversarial network (GAN) is constructed. The deep convolution generative adversarial networks method (DCGAN) is used as the basic architecture of the generator, and the convolutional neural network (CNN) is used as the basic architecture of the discriminator to model and generate the environment of the animation scene. Then, physically-based rendering (PBR) technology is combined to simulate the propagation and reflection of light in the scene for lighting modeling. Percentage closer filtering (PCF) technology is used to process soft shadows so as to enhance the realism of the animation scenes. The Monte Carlo path tracing algorithm is used to comprehensively calculate the global lighting effects of various areas in the scene. Finally, the Tone Mapping algorithm is used for color space conversion and mapping processing, and efficient rendering and realistic effects are demonstrated through deferred rendering and dynamic lighting processing techniques, resulting in the final dynamic scene generation rendering. The experiment is divided into two tasks: dynamic environment modeling and light and shadow effect design and optimization. MT-DCGAN (multi-task deep convolutional generative adversarial network) achieves a design efficiency of 95.73%, with an improvement of 13.49% compared to that of DCGAN, and a user's attention concentration of 93.36%. The combination of multi-task learning algorithms in animation scene design greatly improves design efficiency and user experience, and enhances user satisfaction with animation scenes, thereby contributing an important innovative approach to the animation industry.

Keywords: Metaverse direction; animation scene design; multi-task learning algorithm;*Spirited Away*; immersive experience

1. INTRODUCTION

With the continuous evolution of virtual reality technology and the emerging concept of the metaverse [1–2], animation scene design, as an important part of the metaverse, involves the important task of creating virtual worlds and enriching user experiences. At present, the designing of traditional animation scenes faces many challenges. The traditional single-task processing method has led to poor design efficiency, the lack of effective collaboration and integration between various tasks such as scene construction and animation production, and the insufficient immersive experience of users in the metaverse, making it difficult to achieve interactive experiences comparable to the real world. This study explores the use of multi-task learning algorithms to improve the efficiency and user experience of animation scene design, thereby promoting the development and popularization of the metaverse.

To solve the problems of poor design efficiency and inadequate user immersion in animation scenes, this article focuses on the metaverse and adopts a multi-task learning algorithm-based animation scene design method. In the experiment, a multi-task learning framework based on generative adversarial network (GAN) is constructed, and deep convolution generative adversarial networks (DCGAN) and a convolutional neural network (CNN) are combined for environment

*Corresponding author. Email:14784626413@163.com, 18324552030@163.com, gyu662026@163.com

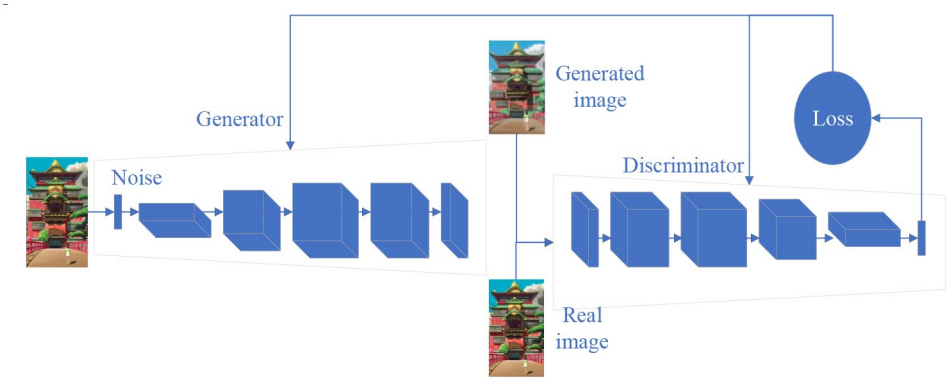


Figure 1 Generation process of GAN scenario.

modeling and generation. Simultaneously, physically-based rendering (PBR) and soft shadow processing technologies are applied to simulate realistic light propagation and reflection effects. In this study, the Monte Carlo path tracing algorithm and Tone Mapping algorithm are utilized to optimize global lighting and color space conversion, ultimately achieving efficient rendering and realistic effects. The experimental results show that the MT-DCGAN (multi-task deep convolutional generative adversarial network) method performs excellently in terms of design efficiency and user immersive experience, significantly improving the quality of animation scene design and user satisfaction.

2. RELATED WORK

Animation scene design is an important part of animation production, which includes the design of background, layout, props, and light and shadow. With the increasing popularity of animation and other fields, many scholars have begun to study animation scene design and have achieved a large number of promising research results. To solve the weight problem of digital sculpture software in animation, scholars such as Zhao adopted computer-aided virtual reality methods to design graphics in 3D animation scenes, providing a solid research basis for animation scene design [3]. Bao applied intelligent algorithms for animation design in 3D graphics engines, ensuring good animation effects while improving frame rates [4]. Ding used the aggregate view object detection (AVOD) algorithm for animation design, which improved the design efficiency. GAN has been widely used to generate realistic characters, transferring the style of reference animation faces to the generated animation faces [5–6]. Reinforcement learning algorithms have shown great potential in optimizing character behavior and actions, improving the design efficiency of animation characters [7–8]. Dynamic knowledge graphs and convolutional neural networks have been applied in scene modeling and animation generation of characters, greatly improving the efficiency of animation generation [9–10]. Although the methods used by these scholars have improved design efficiency in a single-task animation design, they have been unable to achieve efficient collaboration and sharing among multiple tasks, making the overall design process is cumbersome and time-consuming. Hence, comprehensive multi-task learning algorithms have become an important research direction to improve overall design efficiency.

In recent years, the metaverse has emerged, and the experience of animation design has become a key focus of attention. Multi-task learning algorithms, as a machine learning method, can learn multiple related tasks simultaneously. The utilization of shared information can improve the overall performance of models, and many scholars have begun to apply multi-task learning algorithms to improve efficiency. Multi-task learning algorithms are widely used in gradient descent design, quality, and fault prediction modeling, improving the efficiency of modeling and design [11–13]. Ding et al. used task learning algorithms to repair three different types of color-distorted images, greatly improving the recovery efficiency [14]. Scholars such as Hervella utilized multi-task learning algorithms to adaptively optimize the training process of deep learning algorithms, providing a solid foundation for improving design performance [15]. The three-dimensional virtual reality architecture of the metaverse has been used in interaction design in fields such as animation education, providing users with a good interactive experience effect [16–17]. These scholars have combined multi-task learning algorithms and to some extent achieved more efficient collaboration and sharing between different tasks, but have not conducted in-depth research on animation scene design.

3. MULTI-TASK LEARNING ALGORITHMS AND METAVERSE-ORIENTED ANIMATION SCENE DESIGN

3.1 Animation Scene Modeling

3.1.1 Design of GAN Architecture

The environment generation of animation scenes is one of the key steps in the production of animation. In the metaverse-oriented animation scene design, GAN is used for environment generation in this experiment. Generative adversarial network is a deep learning model composed of a generator and discriminator, which mainly generates realistic data samples through adversarial training [18–20]. The process of generating GAN scenarios is shown in Figure 1.

As shown in Figure 1, a similar animation scene is generated through the generator after the original real tower image is input. Then, the discriminator is used to evaluate the similarity between the real image and the generated image, and the loss is calculated. Finally, more realistic animation scene images are continuously optimized and generated.

(1) Design of generator

In GAN, the generator is mainly responsible for generating realistic virtual images, and the discriminator is responsible for distinguishing between the generated virtual images and real images. For the generator, this study adopts DCGAN as the basic architecture of the generator network [21]. The structure of DCGAN includes deconvolution layers, normalization layers, and rectified linear unit (ReLU) activation functions. It is assumed that the input of the generator is a noise vector c with a dimension of d, and the output is an image designed for the animation scene with a size of W × H × T, where W represents the width of the image; H represents the height of the image; T represents the number of channels.

The representation of the generator is shown in Formula (1).

$$y = G(c) \tag{1}$$

where G represents the generator function, and y represents the corresponding output.

In the generator network structure, the input layer receives a noise vector c and then maps the input vector to the intermediate representation A using a fully connected layer. After batch normalization and ReLU activation function processing, the intermediate representation A is transformed into a feature map of the same size as the target image of the animation scene.

The loss function of the generator minimizes the difference between the generated image and the real image, and the optimization objective of the generator is expressed as Formula (2).

$$\min_G V(G, D) = E_{y\sim p_{d(y)}}[\log(1 - D(y))] - E_{c\sim p_{c(c)}}[\log(D(G(c)))] \tag{2}$$

where, $p_d(y)$ represents the distribution of real images; D represents the discriminator function; $p_c(c)$ represents the distribution of random noise vectors.

(2)Design of discriminator

This study adopts CNN as the network structure of the discriminator, which includes convolutional layers, pooling layers, and fully connected layers. The input of the discriminator is image x, and the output is the probability that the image is a real image.

In the network structure of the discriminator, an input layer is used to receive image x, and features of animation scene images are extracted through multiple convolutional layers. After each layer, batch normalization and ReLU activation functions are connected to reduce the size of the feature map through layer-by-layer down-sampling. The output of the convolutional layer is converted into a one-dimensional vector and the result is output through a fully-connected layer and a sigmoid activation function.

The loss function of the discriminator aims to maximize the probability of recognizing real animation scene images as real images. The expression of optimization objectives is shown in Formula (3).

$$\max_D V(G, D) = E_{y\sim p_{d\ (y)}}[\log(D(y))] + E_{c\sim p_{c(c)}}[\log(1 - D(G(c)))] \tag{3}$$

3.1.2 Adversarial Training and Loss Function Design

During adversarial training, the generator and discriminator alternately optimize their respective loss functions, enabling the generator to produce increasingly realistic images and the discriminator to increasingly accurately distinguish between real images and generated images. The steps for adversarial training are as follows.

(1) A batch of real images $y^{(j)}$ are sampled from the collected real data distribution $p_d\ (y)$.

(2) ased on the prior distribution $p_c(c)$, the random noise vector $c^{(j)}$ is sampled.

(3) A batch of images are generated using a generator and the discriminator D is updated. The updated expression is shown in Formula (4).

$$\beta_D \leftarrow \beta_D + \gamma \nabla_{\beta_D} \left(\frac{1}{n} \sum_{j=1}^{n} [\log D(y^{(j)}) + \log(1 - D(G(c^{(j)})))] \right) \tag{4}$$

where, γ represents learning rate, and n represents batch size.

The update expression for generator G is shown in Formula (5).

$$\beta_G \leftarrow \beta_G + \gamma \nabla_{\beta_G} \left(\frac{1}{n} \sum_{j=1}^{n} \log D(G(c^{(j)})) \right) \tag{5}$$

In the design of the loss function, this article applies the Wasserstein Generative Adversarial Network - Gradient Penalty (WGAN-GP) loss function [22–23], which ensures continuity through gradient penalty terms. The loss function of the discriminator is shown in Formula (6).

$$Loss_D = -E_{y\sim p_{d(y)}}[D(y)] + E_{c\sim p_{c(c)}}[D(G(c))] + \delta E_{\hat{y}\sim p_{\hat{y}}}[(||\nabla_{\hat{y}} D(\hat{y})||)_2] \tag{6}$$

where, δ represents the gradient penalty coefficient, and $\hat{y}$ represents the interpolation between the real image data and the generated data.

The loss function of the generator is shown in Formula (7).

$$Loss_G = -E_{c\sim p_c(c)} D(G(c)) \tag{7}$$

3.2 Design of Light and Shadow Effects

3.2.1 Lighting Modeling and Environmental Lighting Processing

In animation scene design, the lighting effect is one of the key factors affecting the realism of the scene. Lighting modeling

uses simulated light sources to showcase the lighting effects of objects in the scene, and environmental lighting processing adds overall brightness and atmosphere to animation scenes by simulating uniform lighting from all directions.

In lighting modeling, PBR technology is used in the experiment to simulate the propagation and reflection process of light in the scene. Formula (8) is used to calculate the Fresnel Formula, which is used to describe the reflectivity of the material surface.

$$B(\zeta_j) = B_0 + (1 - B_0)(1 - cos\zeta_j)^5 \tag{8}$$

where, B_0 represents the Fresnel reflectivity of the material, and ζ_j represents the angle between the incident light and the normal.

After the above steps, this article uses Lambert's law to describe the diffuse reflection component, and its calculation is shown in Formula (9).

$$Q_o(r, \eta_o) = \int \Omega g_s(r, \eta_j, \eta_o) Q_j(r, \eta_j)(m \cdot \eta_j) d\eta_j \tag{9}$$

where, $g_s(r, \eta_j, \eta_o)$ represents the reflectivity function; $Q_o(r, \eta_o)$ represents the intensity of the emitted light; $Q_j(r, \eta_j)$ represents the intensity of incident light; m represents the normal; η_o represents the direction of observation; η_j represents the incident light.

For the processing of environmental lighting, the experiment uses environmental mapping technology to simulate environmental lighting, fully considering the environmental lighting effects in various directions in the animation scene. The environmental mapping technology maps the environmental information in the animation scene to a spherical texture map, and then the corresponding environmental lighting color is obtained from the map based on the observer's position and direction. The environmental mapping is achieved through dynamic updates, which can effectively improve the realism of animation scenes.

3.2.2 Shadow Generation and Soft Shadow Processing

In animation scene design, shadows are one of the important elements that enhance scene realism and three-dimensional perception. To determine the areas that light cannot reach, shadow generation technology is applied in the experiment for processing. Shadow generation is the process of determining shadow regions by determining the occlusion relationship between light rays and objects in the scene. In real-time rendering, intersection detection between light rays and objects is used to determine shadow regions. The expression for shadow generation is in Formula (10).

$$F(O, P) = \prod_{j=1}^{m} R(N(O, P)_j) \tag{10}$$

where, O represents the position of the light source, and $F(O, P)$ represents the visibility from O to P. The value of $R(N(O, P)_j$ is either 0 or 1. If the ray intersects with an object, it is 0; otherwise, it is 1.

To make the shadow edges softer and more realistic, soft shadow processing is now applied to enhance the realism of the animation scene. The percentage closer filtering (PCF) technology [24] is used to process soft shadows in the experiment. The calculation of PCF is shown in Formula (11).

$$F(O) = \frac{1}{M^2} \sum_{j=1}^{M} \sum_{k=1}^{M} R(N(O, P_{jk})) \tag{11}$$

where, M represents the number of sampling points, and P_{jk} represents the points uniformly sampled within the light source area.

3.2.3 Global Lighting Processing

In the metaverse environment of this experiment, for ray tracing, the simulation of the propagation path of light in the scene is used to calculate the energy of light received by each pixel, and realistic light and shadow effects are output. The steps for ray tracing in animation scenes are as follows.

(1) The maximum number of reflections is set and a depth of 0 is defined.

(2) The intersection point between light rays and objects in the scene is calculated.

(3) If there is an intersection, the direct lighting contribution at the intersection is calculated. If the depth is less than the maximum depth, the contribution of reflected and refracted rays is recursively calculated based on the reflectivity and index of refraction of the object surface.

(4) Direct and indirect lighting contributions are merged to output pixel color values.

Global lighting processing is a key technology for achieving realistic lighting effects. In metaverse-oriented animation scene design, global lighting processing calculates that light rays undergo multiple reflections and scattering in the scene, presenting a more realistic lighting model. Global lighting processing is the simulation of the global propagation and interaction process of light in a scene, achieving comprehensive calculation of the global lighting effect of various regions in the scene. This experiment applies Monte Carlo path tracing technology for global light processing. Monte Carlo path tracing [25–27] is calculated with Formula (12):

$$Q_q(r) = Q_o(r) + \int \Omega g_s(r, \eta_j, \eta_q) Q_j(r, \eta_j)(m \cdot \eta_j) d\eta_j \tag{12}$$

where, $Q_o(r)$ represents environmental lighting; $Q_j(r, \eta_j)$ represents the intensity of incident light; η_j represents the incident light.

3.2.4 Color Mapping Processing

In animation scene design, the combination of color and lighting is crucial for improving the realism and visual effects of the scene. In this experiment, the color mapping technology is used to map high dynamic range (HDR) lighting data to low dynamic range (LDR), preserving lighting details and



Figure 2 Framework diagram of multi-task learning.

color information, blending lighting effects with scene colors and presenting a more natural and realistic image.

(1) Color space conversion

In the experiment, the color information in animation scenes is converted from the RGB (red, green, blue) color space to other color spaces, where the color perception of the human eye can be described through the Lab color space. The formula for color space conversion is shown in Formula (13).

$$\begin{cases} Light^* = g(Light) \\ HL^* = g(HL) \\ LH^* = g(LH) \end{cases} \tag{13}$$

where, $Light^*$ represents the brightness of the color space, and HL^* and LH^* represent the red-green component and blue-yellow component, respectively.

(2) Color mapping processing

After color space conversion, HDR lighting data is mapped to the LDR range, preserving lighting details and color information. In the experiment, the Tone Mapping algorithm is used for mapping processing to maintain the contrast and details of the image. The calculation formula for color mapping is shown in Formula (14).

$$Light_{LDR}(y,z) = \frac{Light_{HDR}(y,z)}{1 + Light_{HDR}(y,z)} \tag{14}$$

where, $Light_{LDR}(y,z)$ represents the light intensity within the LDR range, and $Light_{HDR}(y,z)$ represents the light intensity within the HDR range.

3.2.5 Real-Time Light and Shadow Optimization

In animation scene design, to achieve efficient rendering and realistic effects, deferred rendering and dynamic lighting processing technologies are applied for processing. In deferred rendering, the geometric information and lighting information of the scene are first processed separately in the geometric processing stage of the rendering pipeline, and the lighting information is stored in the buffer of the lighting cache. Then, in the post-processing stage, the lighting cache is sampled and calculated to achieve the rendering of lighting effects.

In dynamic scenes, the position and intensity of light sources change over time, and real-time lighting information needs to be updated in animation scenes. During the processing of dynamic lighting, precomputation and caching of lighting data are utilized, and updates are made when changes occur in the scene, reducing real-time computation. This experiment utilizes radiosity maps to achieve realistic lighting and shadow effects in real-time rendering of dynamic scenes. In dynamic lighting processing, the calculation of lighting updates is shown in Formula (15).

$$Light_j(s) = Light_j(s_0) \times \frac{d(s)}{d(s_0)} \tag{15}$$

In Formula (15), $Light_j(s)$ represents the lighting intensity.

3.3 Design of Multi-Task Learning Framework

3.3.1 Design of Multi-Task Learning Architecture

To synchronize multiple tasks such as animation scene modeling and light and shadow effect design, and improve design efficiency and user experience, a multi-task learning framework is designed as shown in Figure 2.

In Figure 2, it can be seen that the experiment adopts a soft parameter multi-task learning method [28–29]. In the multi-task fusion layer, the parameters of each layer are shared, but there are also specific task layer parameter settings, which ensure the independence between tasks while ensuring the sharing of task parameters.

In animation scene design, shared features between different tasks, light-and-shadow effect design, and scene modeling are related. The soft parameter sharing method helps the model learn correlations better, improving design efficiency and quality.

In multi-task learning frameworks, shared feature extractors are primarily used to extract shared feature representations from input data, utilizing CNN to effectively learn abstract representations of input data, including the geometric structure of the scene and lighting information. Task-specific network language handles specific information for scene modeling and lighting effect design tasks. DCGAN and CNN are mainly used for modeling animation scenes, and technologies such as Monte Carlo path tracing algorithm are used for the design of light and shadow effects. The multi-task fusion layer effectively combines shared feature extractors with task-specific networks to output the final learning results.

3.3.2 Design of Parameter Sharing

Parameter sharing is a key design strategy that fully utilizes the correlation between different tasks, reduces the number of model parameters, reduces model complexity, and improves model training efficiency and generalization ability. Parameter sharing can be divided into weight sharing, feature mapping sharing, and dynamic parameter sharing.

In the sharing layer, key weight parameters are set so that they are shared among different tasks. In the experiment, parameter sharing is done by means of feature mapping sharing,, whereby a portion of the feature mapping is shared with the input data of the task-specific network, so that different tasks can share some feature representations. The correlation between different tasks changes as the training process progresses. To more flexibly adapt to the correlation between different tasks, a dynamic parameter-sharing mechanism is designed to adjust the degree of parameter sharing in real time.

3.3.3 Design of Loss Function

In the design of animation scenes for the metaverse, this experiment uses a comprehensive loss function to determine the effectiveness of multi-task learning. The loss function of each task is represented as an independent term and combined into a comprehensive loss function for optimization. The multi-task loss function is shown in Formula (16).

$$Loss_m = \sum_{j=1}^{n} \zeta_1 \cdot Loss_j \tag{16}$$

where, $Loss_m$ represents the multi-task loss function, and ζ_1 represents the weight of the task.

In animation scene modeling, combining Formulas (6) and (7) are used to obtain the discriminator and generator losses in animation scene modeling and generation. In the task of designing light and shadow effects, the experiment uses a pixel-level loss function to measure the difference between the generated image and the real image, as shown in Formula (17).

$$Loss_g = \frac{1}{n} \sum_{j=1}^{n} ||I_g(j) - I_r(j)||_2^2 \tag{17}$$

where, I_g represents the generated image; I_r represents the real image; n is the number of pixels in the image.

4. EXPERIMENT AND VERIFICATION OF ANIMATION SCENE DESIGN

4.1 Experimental Environment

Hardware environment: Intel Core i9-10900K CPU (central processing unit), NVIDIA GeForce RTX 3090 GPU (graphics processing unit), 64GB DDR4 RAM (random access memory), 1TB SSD (solid state drive).

Software environment: Windows 10, TensorFlow 2.5.0 framework, OpenCV 4.5.3, NumPy 1.21.0, Jupyter Notebook.

4.2 Experimental Data and Preprocessing

(1) Experimental data

The data for this experiment comes from the animation *Spirited Away*, which is a classic animation film directed by Hayao Miyazaki. The animation includes exquisite scene design and imaginative worldview, covering a rich range of scenes. A total of 2538 images are collected for the experiment, including various types of images such as streets, forests, sea, ancient villages, natural and artificial buildings. The 10-fold cross-validation method is used in the experiment to divide the training and testing sets. The experiment is conducted in several rounds and the average of the results is taken as the final evaluation score.

In Figure 3, it can be seen that the collected data includes streets, forests, sea, ancient villages, and artificial buildings, providing sufficient types of animation scenes for the experiment.

(2) Data preprocessing

The experimental preprocessing requires image size adjustment and data augmentation. The image size is uniformly set to 1722*1053, and reverse selection of 30 degrees and random cropping processing are used for data augmentation. The preprocessing results are shown in Figure 4.

In Figure 4, the first row is the rotated images, and the second row is the randomly cropped images, ensuring the richness of the dataset.

4.3 Experimental Design

The experimental design steps are as follows:

(1) Firstly, images are collected from the animation *Spirited Away*, and the size adjustment and data augmentation are performed on the collected images.



Figure 3 Some experimental data.



Figure 4 Preprocessed results.

(2) A multi-task learning framework is constructed, and the tasks are divided into dynamic environment modeling and light-and-shadow effect design and optimization.

(3) In dynamic environment modeling, DCGAN is used as the basic architecture of the generator, and CNN is used as the basic architecture of the discriminator. Then the environment is modeled and generated for the animation scene.

(4) For the design and optimization of light and shadow effects, experiments are conducted using PBR technology to simulate the propagation and reflection process of light in the scene, and lighting modeling is carried out. Environmental mapping technology is used to simulate environmental lighting for subsequent processing.

(5) The intersection detection of rays and objects is used to determine the shadow area of the image, and then the PCF technology is used to process soft shadows to enhance the realism of the animation scene. The Monte Carlo path tracing algorithm is used for the comprehensive calculation of global lighting effects for various areas of the scene.

(6) The Tone Mapping algorithm is integrated for color space conversion and mapping processing, and deferred rendering and dynamic lighting processing technologies are applied to achieve efficient rendering and realistic effects.

(7) The final dynamic scene generation rendering of the experiment is output, and its scene generation performance and effect are analyzed.

The hyperparameter settings are shown in Table 1.

4.4 Evaluation indicators

Peak signal-to-noise ratio (PSNR);

$$PSNR = 20 \cdot log_{10}(\frac{MAX_J}{\sqrt{MSE}}) \tag{18}$$

where, MAX_J represents the maximum value of image data.

Structural similarity (SSIM):

$$SSIM(x, y) = \frac{(2\rho_x\rho_y + c_1)(2\nu_{xy} + c_2)}{(\rho_x^2 + \rho_y^2 + c_1)(\nu_x^2 + \nu_y^2 + c_2)} \tag{19}$$

Table 1 Hyperparameter settings.

Parameter	Value	Parameter	Value	Parameter	Value
Learning rate	0.001	Number of generator layers	4	Discriminator loss function weight	1.0
Batch size	64	Number of discriminator layers	4	Gradient penalty coefficient	10.0
Number of training epochs	100	Weight decay coefficient	0.0001	Number of PCF sampling points	16
Noise vector dimensions	100	Generator loss function weight	1.0	Maximum number of reflections	5

where, ρ_x and ρ_y represent the mean of the image; ρ_x^2 and ρ_y^2 represent variance; ν_{xy} represents covariance; c_1 and c_2 represent constants.

Frechet inception distance (FID):

$$FID = \parallel \rho_r - \rho_g \parallel_2^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}) \quad (20)$$

Difference structural similarity (DSSIM):

$$Dssim = \frac{1 - SSIM}{2} \quad (21)$$

Mean absolute error (MAE):

$$MAE = \frac{1}{WH} \sum_{i=1}^{W} \sum_{j=1}^{H} |J_r(i, j) - J_g(i, j)| \quad (22)$$

where, W represents the width of the image, and H represents the height of the image.

Spectral condition index (SCI):

$$SCI = \frac{2 \times |S_r \cap S_g|}{|S_r| + |S_g|} \quad (23)$$

where, S_r represents the shadow area of the real image, and S_g represents the shadow area of the generated image.

Illumination smoothness (IS):

$$IS = \frac{1}{WH} \sum_{i=1}^{W} \sum_{j=1}^{H} (\nabla J_g(i, j) - \nabla J_r(i, j))^2 \quad (24)$$

where, ∇J represents the gradient of the image.

4.5 Experimental results

(1) Results of experimental generation

The generated results of animation scene design are shown in Table 2, which presents the original images, types, and generated animation scene images in sequence.

In Table 2, it can be seen that for the artificial building numbered 1, there is a significant difference between the original image and the generated image. In the generated image, the sky color changes and the floors are grayed out, presenting a more ancient atmosphere compared to the original image. Image number 2 is an ancient village, and here there is also a significant difference between the original image and the generated image, mainly manifested in the sky. The sky in the generated image is a darker blue, and the road surface is more yellowish, which greatly enhances the colorthe animation scene at this time. The generated images numbered 3 to 5 show significant differences from the original images, resulting in a more diverse animation scene.

(2) Comparison of performance of animation scene generation of different models

To analyze the performance of the animation scene generation of the MT-DCGAN model in this experiment, it is compared and analyzed with DCGAN, Cycle-Consistent Generative Adversarial Networks (CycleGAN), GAN, and Pix2Pix. The performance of generating animation scenes for different models is shown in Figure 5. In Figure 5, Figure 5(a) represents the PSNR and FID data of different models during animation scene generation, and Figure 5(b) represents the DSSIM and SSIM data of different models during animation scene generation.

In Figure 5(a), the higher the PSNR, the higher is the similarity between the generated image and the real image, resulting in better image quality. The PSNR of MT-DCGAN reaches 30.52dB, significantly higher than that of other models; the PSNR of DCGAN and Pix2Pix are 27.17dB and 28.63dB, respectively; the PSNR of CycleGAN and GAN are 24.31dB and 22.72dB respectively, resulting in poor similarity of the generated images. The FID of MT-DCGAN is 25.42, which is significantly better than other models, indicating that the generated images are visually closer to real images; the FID of Pix2Pix is 40.15, second to MT-DCGAN; the FID values of DCGAN, CycleGAN, and GAN are 55.65, 75.02, and 90.33, respectively, indicating poor image quality.

Figure 5(b), shows that the SSIM of MT-DCGAN is 0.95, indicating that its generated image is structurally very similar to the real image. The SSIM of Pix2Pix and DCGAN are 0.92 and 0.88, respectively. The DSSIM of MT-DCGAN is only 0.05, indicating that its generated image has the smallest structural deviation from the real image. The DSSIM of Pix2Pix and DCGAN are 0.08 and 0.12, respectively.

Overall, it can be concluded that MT-DCGAN exhibits the best performance in generating animation scenes among the different models. Because it combines the advantages of multi-task learning, it can simultaneously optimize image generation and light and shadow effect design, improving the overall quality of generated images. Additionally, MT-DCGAN helps to capture the details and complex features of images, improving the clarity and quality of generated images.

(3) Comparison of performance in light and shadow effect design

Table 2 Generation effect of animation scenes.

Number	Original image	Type	Generated image
1		Artificial building	
2		Ancient village	
3		Sea	
4		Artificial building	
5		Sea	

There are various types of animation scenes in *Spirited Away*. It contains streets, forests, sea, ancient villages, natural and artificial buildings, which are analyzed in this study. The effect of light and shadow design is shown in Figure 6. Figure 6(a) shows the indicator data of MAE and mean squared error (MSE) for different scenes, and Figure 6(b) shows the indicator data for SCI and IS in different scenes.

In Figure 6, it can be seen that there are significant differences in the performance of different scenes in regard to generating light and shadow effects. The MAE of the generated streets is the lowest at only 0.034, and its light and shadow effect is closest to the real scene; the MSE is 0.009; the SCI is 0.95, indicating the generated light and shadow effects are most consistent with the real scene in terms of color information; the IS is 8.5, and the generated light and shadow effects have the best perceptual quality.

The MAE of forest scene generation is 0.089, indicating the maximum deviation between the generated light and shadow effects and the real scene; the MSE reaches 0.025; the SCI is 0.82; the IS is 6.0. The light and shadow design effects of the sea, ancient village, natural building, and artificial building animation scenes are all less than those of the streets, but they also achieve good results.

In summary, it can be seen that the generation of street scenes performs best due to their simple geometric structure and stable light sources, while the generation of forest scenes

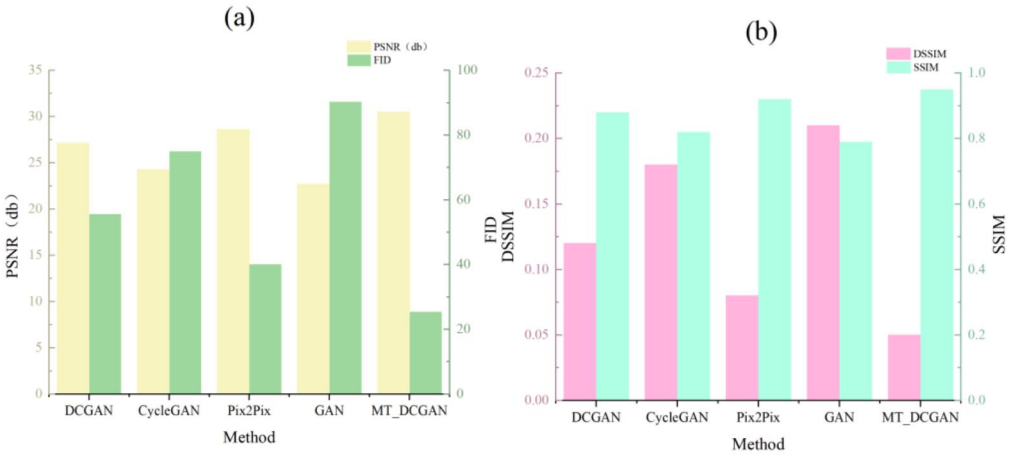


Figure 5 Performance of animation scene generation of different models.

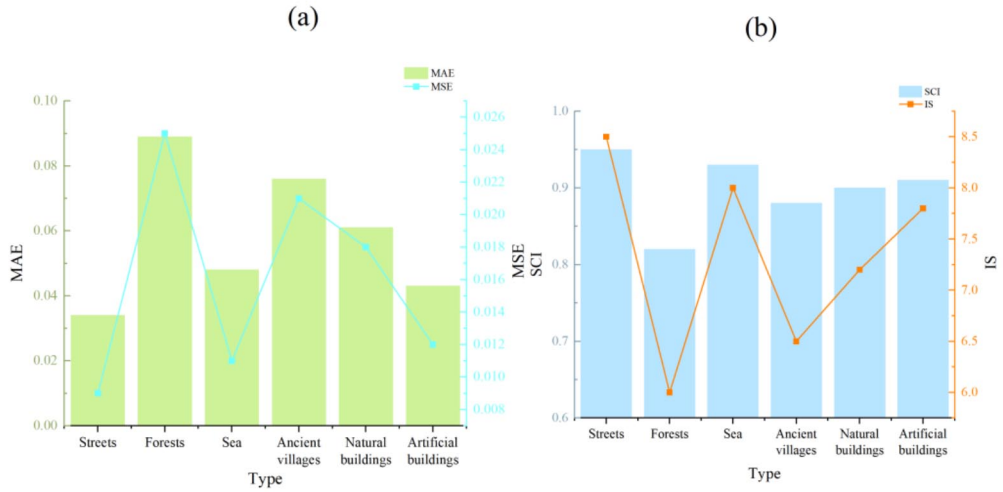


Figure 6 Light and shadow design effect.

performs worst due to their complex geometric structure and dynamic light and shadow changes. The reflection and refraction effects on the water surface in the sea scene are complex, making it difficult to accurately simulate the propagation and changes of light on the water surface. Moreover, the fluctuations and changes in light and shadow on the water surface increase the difficulty of generating light and shadow effects. Ancient village, natural building, and artificial building scenes are usually composed of different building materials, and the diversity of their light and shadow effects also increases the complexity of generation.

(4) Design efficiency and time performance of animation scenes of different models

To verify the effectiveness of the multi-task learning algorithm, it is compared with the algorithm model of a single task, and the design efficiency and time performance of the animation scene are statistically analyzed, as shown in Figure 7. In Figure 7, Figure 7(a) shows the design efficiency and resource utilization of different models, and Figure 7(b) gives the response time and convergence speed of different models.

From Figure 7, it can be seen that the design efficiency of DCGAN is 82.24%, and that of MT-DCGAN is 95.73%, which is an improvement of 13.49% compared to that of DCGAN. It can be seen that the improvement effect is significant, and the application of multi-task learning is effective and feasible, greatly improving design efficiency. The design efficiency of Pix2Pix is 86.81%, and the design efficiency of CycleGAN and GAN are 77.57% and 69.36%, respectively, indicating poor performance. The resource utilization rate of MT-DCGAN is 88.21%, showing the best performance. The resource utilization rates of Pix2Pix and DCGAN are 84.92% and 81.09%, respectively. The resource utilization rates of CycleGAN and GAN are 75.86% and 73.75%, respectively.

In terms of time performance, this experiment analyzes this dimension from two aspects: response time and convergence speed. MT-DCGAN has the fastest response time for generating images, being only 2.87 seconds per image, and Pix2Pix has a response time of 3.12 seconds per image. MT-DCGAN has the fastest convergence speed, at 162 epochs. The convergence speed of CycleGAN and GAN is 201 and

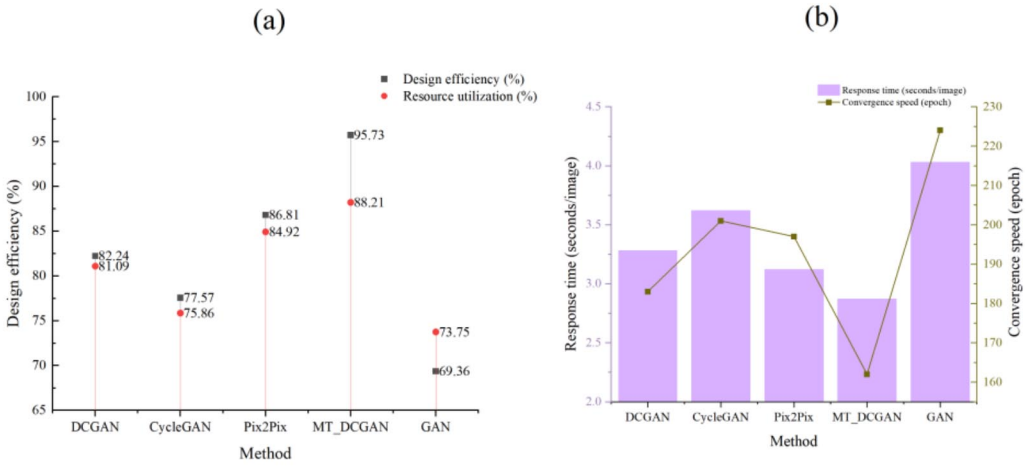


Figure 7 Animation scene design efficiency and time performance of different models.

Table 3 Evaluation data of users' immersive experience.

Method	Attention concentration (%)	Stay time (s)	Emotional experience index
MT_DCGAN	93.36	21.84	85.45
DCGAN	83.78	18.57	76.52
CycleGAN	78.43	16.73	70.59
GAN	75.92	15.29	72.14
Pix2Pix	87.21	20.01	80.93

224 epochs respectively, requiring more training time to reach the convergence state compared to MT-DCGAN.

From the above, it can be seen that MT-DCGAN performs the best in terms of animation scene design efficiency and time performance. Its high design efficiency and resource utilization, and fast response time and convergence speed demonstrate the significant advantages of multi-task learning algorithms in animation scene design.

(5) Users' immersive experience

The immersive experience of users is of great significance in animation scene design, as it directly affects user satisfaction, scene design quality, and user interaction participation. This experiment collects data from 3215 users through a questionnaire survey, with the user group being young people who enjoy watching animations. In terms of users' immersive experience, this study quantifies the level of attention concentration, stay time, and emotional experience index. The evaluation data of users' immersive experience is shown in Table 3.

In Table 3, MT-DCGAN has the highest level of attention concentration of 93.36%, which is much higher than that of other models. It can be seen that the animation scenes generated by MT-DCGAN can effectively attract users' attention. The attention concentration of Pix2Pix reaches 87.21%, and the attention concentration of DCGAN, CycleGAN, and GAN are 83.78%, 78.43%, and 75.92%, respectively. The average stay time of MT-DCGAN is the longest, reaching 21.84 seconds, indicating that users have a longer stay time and a derive a more satisfying experience from the animation scenes generated using MT-DCGAN. The emotional experience index of MT-DCGAN is the highest, at 85.45, indicating that its generated animation scenes can trigger more positive emotional reactions from users.

To further evaluate the users' immersive experience, satisfaction ratings for the visual effects, realism, and sense of time loss of the generated animation are collected, with a total score of 30 points, each direction accounting for 10 points. The satisfaction level of users' immersive experience is shown in Figure 8. The satisfaction score of MT-DCGAN reaches 26 points, accounting for 86.7% of the total score. The satisfaction score of Pix2Pix is 24 points, and that of DCGAN is only 22 points. The lowest satisfaction score is that of the traditional GAN model, which is only 16 points, accounting for 53.3% of the total score. Overall, it can be seen that the animation scene effects generated by the MT-DCGAN model are more favored by users.

5. EXPERIMENTAL DISCUSSION

This experiment compares the performance of animation scene designs of different models and determines the effectiveness of the multi-task learning algorithm: MT-DCGAN. In terms of generating animation scenes, MT-DCGAN performs best according to indicators such as PSNR, FID, SSIM, and DSSIM, and the generated images are closer to real images in terms of structure and visual effects. In terms of light and shadow effect design, MT-DCGAN performs well in various scenes such as streets, forests, sea, ancient villages, natural and artificial buildings, and generates more realistic and consistent light and shadow effects. In regard to design efficiency and time performance, MT-DCGAN

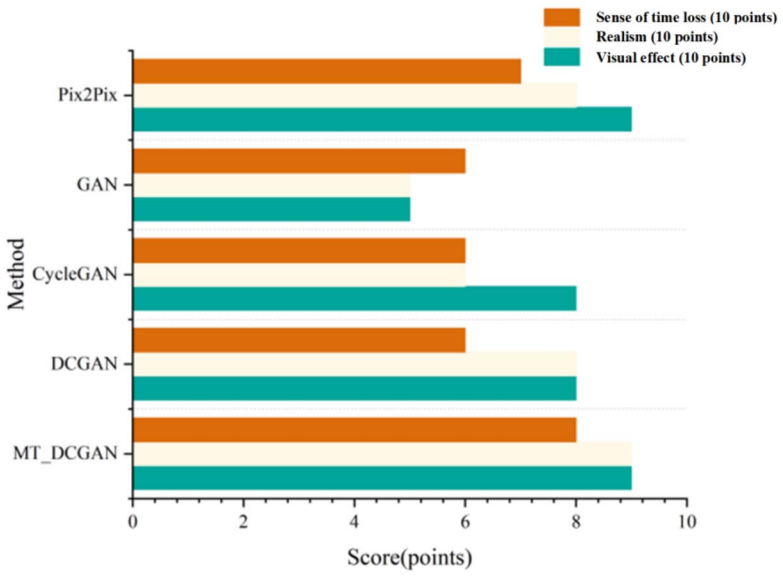


Figure 8 Satisfaction of users' immersive experience.

significantly outperforms other single-task models with the highest design efficiency and resource utilization, and the fastest response time and convergence speed. A questionnaire survey of 3215 users verified that MT-DCGAN performs the best in terms of users' attention concentration, stay time, and emotional experience index, demonstrating its advantages in users' immersive experience.

This experiment demonstrates that the multi-task learning algorithm MT-DCGAN has significant advantages in animation scene design, not only improving the quality of generated images, but also optimizing lighting effects and overall design efficiency. By combining multiple tasks for joint optimization, finely-detailed and more realistic animation scenes can be generated through MT-DCGAN, enhancing the users' immersive experience. The results of this research provide a new technological path for animation production, which helps to improve the efficiency and quality of animation scene design, and enhance users' interactive participation and satisfaction, and provides a significant foundation for future research on animation production and related field.

6. CONCLUSION

This study focuses on the animation *Spirited Away* and uses multi-task learning algorithms to study the design of animation scenes. In the experiment, a multi-task learning framework based on GAN is constructed to model and generate the environment of animation scenes, and combined with PBR and other technologies, the final rendering of the generated dynamic scene is output. The experimental results show that the multi-task learning algorithm performs well in terms of animation scene design and user experience, and the generated animation scene images have excellent realistic effects. Although this study has achieved some noteworthy results, it still has several shortcomings. This experiment is mainly based on the scene of the animation *Spirited Away*, and the diversity and scale of the dataset are relatively limited. Different types of animation scenes have different characteristics and challenges, and the current experimental results cannot fully be applied to the generation effects of other types of animation scenes. Moreover, the complex network architecture and optimization process of multi-task learning algorithms require high computational resources. In future research, the dataset can be expanded to cover more types and styles of animation scenes to validate the effectiveness of the model in different application scenarios. The application of this method in virtual reality and game design can also be explored, and more efficient model optimization methods and lightweight network structures can be studied to reduce computational costs and improve applicability in resource-limited environments.

REFERENCES

1. Wang H, Ning H, Lin Y, Wang W, Dhelim S,Farha F,et al. A survey on the metaverse: The state-of-the-art, technologies, applications, and challenges. IEEE Internet of Things Journal, 2023, 10(16): 14671–14688.
2. Alabau A, Fabra L, Marti-Teston A, Munoz A,Solanes J E,Gracia L. Enriching User-Visitor Experiences in Digital Museology: Combining Social and Virtual Interaction within a Metaverse Environment. Applied Sciences, 2024, 14(9): 3769.
3. Zhao J, Zhao X. Computer-aided graphic design for virtual reality-oriented 3D animation scenes. Computer-Aided Design and Applications, 2022, 19(S5):65–76.
4. Bao W. The application of intelligent algorithms in the animation design of 3D graphics engines. International Journal of Gaming and Computer-Mediated Simulations (IJGCMS), 2021, 13(2): 1–12.
5. Li B, Zhu Y, Wang Y, Lin C W,Ghanem B,Shen L. AniGAN: Style-guided generative adversarial networks for unsupervised anime face generation. IEEE Transactions on Multimedia, 2022, 24: 4077–4091.
6. Li, J. High-precision Motion Capture System Based on 3D Space Animation Scene Construction Based on Optical Fiber Communication Network. Engineering Intelligent Systems, 2025, 33(5): 545–553.

7. Yang Z, Yin Z. Efficient hyperparameter optimization for physics-based character animation. Proceedings of the ACM on Computer Graphics and Interactive Techniques, 2021, 4(1): 1–19.
8. Peng X B, Ma Z, Abbeel P, Levine S,Kanazawa A. AMP: Adversarial motion priors for stylized physics-based character control. ACM Transactions on Graphics (TOG), 2021, 40(4): 1–20.
9. Wang Y. 3D Dynamic Image Modeling Based on Machine Learning in Film and Television Animation. Journal of Multimedia Information System, 2023, 10(1): 69–78.
10. Song W, Zhang X, Guo Y, Li S,Hao A,Qin H. Automatic Generation of 3D Scene Animation Based on Dynamic Knowledge Graphs and Contextual Encoding. International Journal of Computer Vision, 2023, 131: 2816–2844.
11. Zhou X, Gao Y, Li C, Huang Z. A multiple gradient descent design for multi-task learning on edge computing: Multi-objective machine learning approach. IEEE Transactions on Network Science and Engineering, 2022, 9(1): 121–133.
12. Yan H, Sergin N D, Brenneman W A, Lange S J,Ba S. Deep multistage multi-task learning for quality prediction of multi-stage manufacturing systems. Journal of Quality Technology, 2021, 53: 526–544.
13. Xie Z, Chen J, Feng Y, Zhang K,Zhou Z. End to end multi-task learning with attention for multi-objective fault diagnosis under small sample. Journal of Manufacturing Systems, 2022, 62: 301–316.
14. Ding Y, Wu K. A Multi-Task Learning and Knowledge Selection Strategy for Environment-Induced Color-Distorted Image Restoration. Applied Sciences, 2024, 14(5): 1836.
15. Hervella A S, Rouco J, Novo J, Ortega M. Multi-Adaptive Optimization for multi-task learning with deep neural networks. Neural Networks, 2024, 170: 254–265.
16. Zhao Y, Jiang J, Chen Y, Liu R,Yang Y,Xue X,et al. Metaverse: Perspectives from graphics, interactions and visualization. Visual Informatics, 2022, 6(1): 56–67.
17. Lo S C, Tsai H H. Design of 3D virtual reality in the metaverse for environmental conservation education based on cognitive theory. Sensors, 2022, 22(21): 8329.
18. Liu G, Chen X, Gao Z. A Novel Double-Tail Generative Adversarial Network for Fast Photo Animation. IEICE TRANSACTIONS on Information and Systems, 2024, 107(1): 72–82.
19. Aldausari N, Sowmya A, Marcus N, Mohammadi G. Video generative adversarial networks: a review. ACM Computing Surveys (CSUR), 2022, 55(2): 1–25.
20. Yang, J. Detail Enhancement of 3D Animation Images Based on Swarm Intelligence Algorithm. Engineering Intelligent Systems, 2026, 34(2):211–219.
21. Alqahtani H, Kavakli-Thorne M, Kumar G. Applications of generative adversarial networks (gans): An updated review. Archives of Computational Methods in Engineering, 2021, 28: 525–552.
22. Abu-Srhan A, Abushariah M A M, Al-Kadi O S. The effect of loss function on conditional generative adversarial networks. Journal of King Saud University-Computer and Information Sciences, 2022, 34(9): 6977–6988.
23. Huang G, Jafari A H. Enhanced balancing GAN: Minority-class image generation. Neural computing and applications, 2023, 35: 5145–5154.
24. Elazab N, El-Sappagh S, Atwan A, Soliman H,Elmogy M,Alarabi L,et al. Overlapping Shadow Rendering for Outdoor Augmented Reality. Computers, Materials & Continua, 2021, 67(2):1915–1932.
25. Zhang C, Yu Z, Zhao S. Path-space differentiable rendering of participating media. ACM Transactions on Graphics (TOG), 2021, 40(4): 1–15.
26. Goswami P. A survey of modeling, rendering and animation of clouds in computer graphics. The Visual Computer, 2021, 37: 1931–1948.
27. Lin Y. The Review of Modeling and Rendering 3D Environment: A Survey. Highlights in Science, Engineering and Technology, 2024, 93: 304–315.
28. Wang H, Jin X, Du Y, Zhang N,Hao H. Adaptive Hard Parameter Sharing Method Based on Multi-Task Deep Learning. Mathematics, 2023, 11(22): 4639.
29. Duan Y, Wang K. Adaptive and robust multi-task learning. The Annals of Statistics, 2023, 51(5): 2015–2039.