Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 9
AIDML 2025

Analysis and Application Research of Typical Technologies of Generative AI in the Context of Games

Shuzhao Shen *

College of Computer and Information Sciences, Southwest University, Chongqing, China

* Corresponding Author Email: limy@tzc.edu.cn

Abstract. With the rapid development of artificial intelligence technology, generative AI technology is gradually reshaping the paradigm of game development. It not only greatly improves the efficiency of game content generation, but also enhances the interactivity and personalized experience of games, which will be a unique new experience for developers and players. This paper focuses on the typical technical architecture and innovative applications of generative AI in the game field, and systematically analyzes the core principles, technical advantages and limitations of typical technologies such as GANs and Diffusion Models. It also derives innovative technologies in this field of generative AI in recent years, such as the PANGeA framework and the Text-to-Game engine. Through innovative technical research, this paper deeply explores the application efficiency of generative AI in scenarios such as character generation, dynamic narrative and code automation, and combines industrial practices such as Inworld AI Character Engine and Roblox's generative tool chain to reveal its potential in improving development efficiency and player immersion. This paper further points out that generative AI still faces challenges in terms of computing resource consumption, content controllability, ethical risks and cross-platform compatibility. In the future, with algorithm optimization and hardware iteration, generative AI is expected to drive the evolution of the gaming industry towards intelligence and personalization through adaptive content generation and multimodal interaction technology.

Keywords: Generative artificial intelligence; game development; programmatic content generation; dynamic narrative; human-computer interaction.

1. Introduction

Traditional game development is highly dependent on a large number of manual designs and scripted processes, and there are bottlenecks such as content homogeneity and long development cycles. Generative AI uses deep learning models (such as GANs, Diffusion Models) and other technologies to autonomously generate high-quality game scenes, characters, tasks and dialogue systems, thereby enhancing the immersion, content richness and personalized experience of the game. The most eye-catching one is the programmatic content generation technology, which allows the game world to change dynamically and improves the playability of the game. Take Minecraft and No Man's Sky as examples. These two games use AI technology to achieve large-scale scene generation, providing players with a virtual world full of infinite possibilities. This technology not only enhances the exploratory nature of the game, but also brings players a unique personalized experience. In addition, the application of AI in NPC (non-player character) intelligent interaction is also increasing. Using reinforcement learning and natural language processing technology, NPC can react in real time according to player behavior to form a more natural game experience.

In recent years, academia and industry have conducted in-depth research on the application of generative AI in the field of game development. Among them, a 2023 research review analyzed the application of GPT models in games from 2020 to 2023, and found that GPT performed well in procedural content generation, interactive game design, and automatic task generation, but still faced problems with content controllability and generation stability [1]. On the other hand, the application of AI in NPC intelligent interaction has also become a research hotspot. For example, a 2023 study on video synthesis and motion control proposed a generative video synthesis method called

VideoComposer, which uses convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to enable game NPCs to learn the movements of player characters and extract their features, thereby synthesizing coherent motion sequences, providing players with higher playability and immersion [2]. Similarly, Diffusion Models has also made breakthroughs in the quality of game character generation. A study last year focused on diffusion models in the field of video generation, making the generated content more realistic and coherent through deep neural networks and the increase or decrease of noise parameters [3]. In addition, a study in early 2024 explored a dynamic plot generation method that combines natural language processing and AI, enabling the game world to automatically adjust the plot development according to player decisions, thereby achieving personalized narrative [4]. The subsequent chapters of this article will conduct an in-depth study and analysis of the epitome of innovative technologies in recent years.

Current research results show that generative AI is gradually building a complete AI empowerment system from content generation to intelligent interaction, bringing profound changes to the game industry. However, developers still face many challenges, among which the more prominent issues are how to ensure the controllability of AI-generated content, how to optimize computing resources to reduce development costs, and how to further improve the creativity and diversity of AI-generated content. With the continuous advancement of hardware upgrades and algorithm optimization, generative AI is expected to play a greater role in personalized game experience, PCG optimization, NPC interaction intelligence, etc. in the future, and become an indispensable innovation engine for the game industry.

2. Technical Analysis

2.1. Typical Theoretical Overview

The application of generative AI technology in game development has gradually become an important force to promote industry innovation. This chapter will study five typical generative AI technologies, namely generative adversarial networks, diffusion models, variational autoencoders, transformer models and large language models, which have played an important role in game development.

2.1.1. Generative Adversarial Networks (GANs)

Generative adversarial networks (GANs) achieve data distribution fitting through adversarial training of generators and discriminators, thereby generating high-quality images, videos, audio and other content, as shown in Figure 1. GANs are widely used in 3D character generation, scene design and texture creation in games due to their high fidelity of generated content, effectively improving the efficiency of art resource generation. A typical case is MetaHuman Creator of Epic Games, as shown in Figure 2. This tool uses GANs technology to generate realistic virtual characters, which is widely used in games and virtual reality, significantly improving the efficiency and realism of character modeling. This is a testament to the strengths of GANs: excellent generation quality, the ability to generate high-quality outputs that often surpass human-generated images in terms of realism and detail [5]. As pointed out by Zhengwei Wang et al. in their study, GANs have made revolutionary progress in image generation tasks, generating high-quality images that even surpass human-generated images in terms of realism [5]. This breakthrough has led to the widespread use of GANs in game character generation and scene design. However, there are also some challenges in the application of GANs, especially during training, where the adversarial nature between the generator and the discriminator makes the training process very unstable. One of the disadvantages is that it is difficult to control and unstable. A common challenge when training GANs is the "mode collapse" phenomenon. This phenomenon manifests itself as the generator tends to generate a limited number of outputs when faced with different inputs, resulting in a significant reduction in the diversity of generated content [5]. As pointed out in the study, the training process of GANs is inherently unstable, mainly due to the adversarial nature between the generator and the discriminator. The generator may fall into a "comfort zone" where the generated output tends to be single even if the input data is diverse. This phenomenon is called mode collapse. This not only limits the diversity of generated content, but also poses challenges to the practicality of the model. Therefore, although GANs perform well in terms of generation quality, the instability and control difficulty of its training process are still urgent problems to be solved.

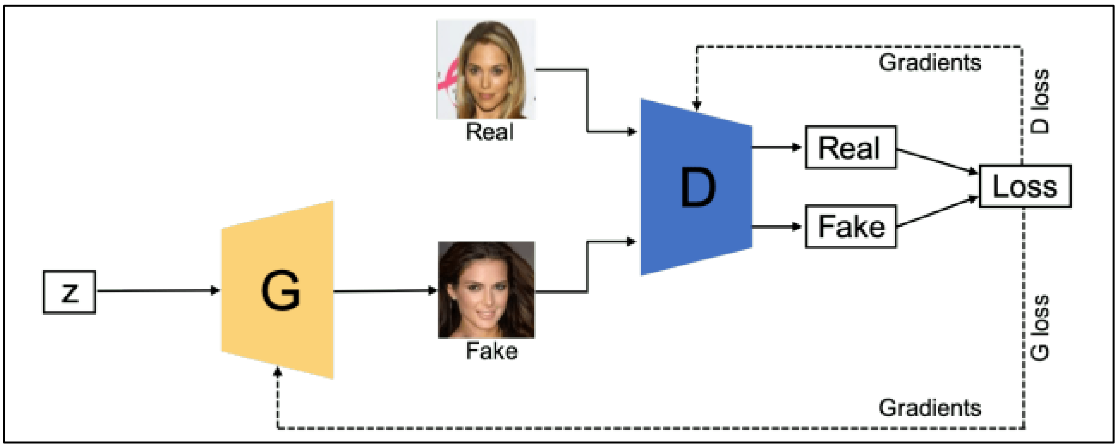

Figure 1. Architecture of a GAN [5].

Figure 2. Epic Games' MetaHuman Creator, a tool that uses GANs to generate realistic and customizable virtual humans.

2.1.2. Diffusion Model

The diffusion model adds noise to the data and reconstructs the original data through an inverse denoising process, as shown in Figure 3. This method has shown excellent results in image generation and style transfer. In game development, the main application of the diffusion model is to generate artistic style images and high-quality game textures. For example, the well-known Stable Diffusion model has been widely used in the generation of game scenes and characters, which can create more realistic and artistic visual content, significantly improving the visual expression and artistic atmosphere of the game. Studies have shown that the diffusion model is superior to the GANs in terms of detail richness when generating game scenes [6]. As pointed out by Prafulla Dhariwal and Alex Nichol, the diffusion model performs well in image generation quality and can generate images with higher fidelity and more details, especially in artistic style creation [6]. However, the diffusion model also has some limitations, the most notable of which is its high demand for computing resources and slow generation speed. Although the diffusion model can generate high-quality images, its iterative generation mechanism leads to a significant increase in computing overhead, which makes it difficult to meet the requirements of real-time rendering. Research shows that the generation process of the diffusion model is more time-consuming and requires more computing resources than GANs. Therefore, although the diffusion model has obvious advantages in image quality, its slow generation speed and high computing cost are still a challenge to be solved in practical applications.

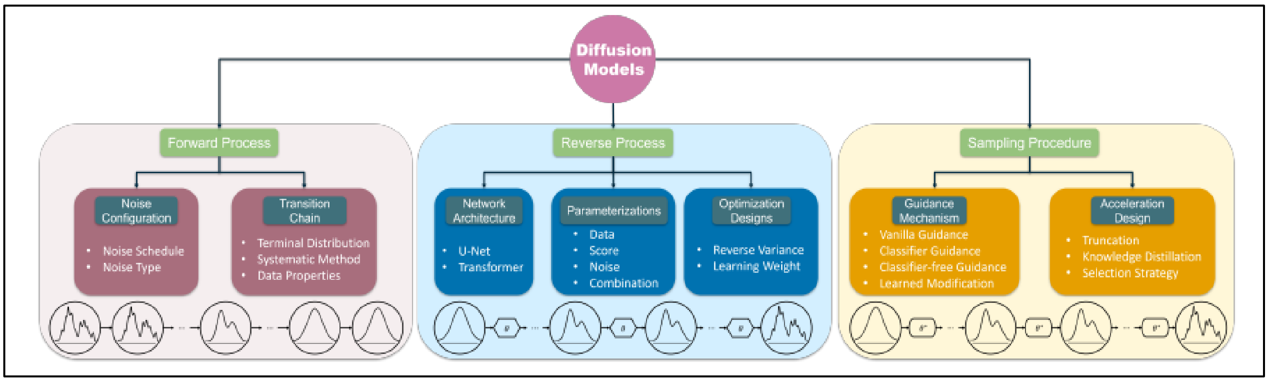

Figure 3. The overview of diffusion models [7].

2.1.3. Variational Autoencoder (VAE)

Variational Autoencoder (VAE) generates data by mapping input data to a latent space and reconstructing the data using a decoder, as shown in Figure 4. An important advantage of VAE is the stability of its training process, and its ability to manipulate latent variables in the latent space to achieve fine control over the generation process. As pointed out by Diederik P Kingma and Max Welling, VAE not only provides a stable training process, but also allows developers to easily manipulate the latent space, thereby more effectively controlling the generated content [8]. This feature has made VAE widely used in many scenarios that require stable training and precise control of generated content. However, VAE also has some limitations, especially in terms of the quality and details of generated images. Although VAE performs well in the training process, the images it generates often lack details and appear blurry. Compared with GANs, the images generated by VAE are slightly inferior in visual quality. As Kingma and Welling's study highlights, despite VAE's advantage in training stability, the images it generates are generally inferior to those generated by GANs in terms of clarity and detail [8]. Therefore, while VAE excels in stability and controllability, its lack of image quality remains a significant challenge when generating high-precision game assets.

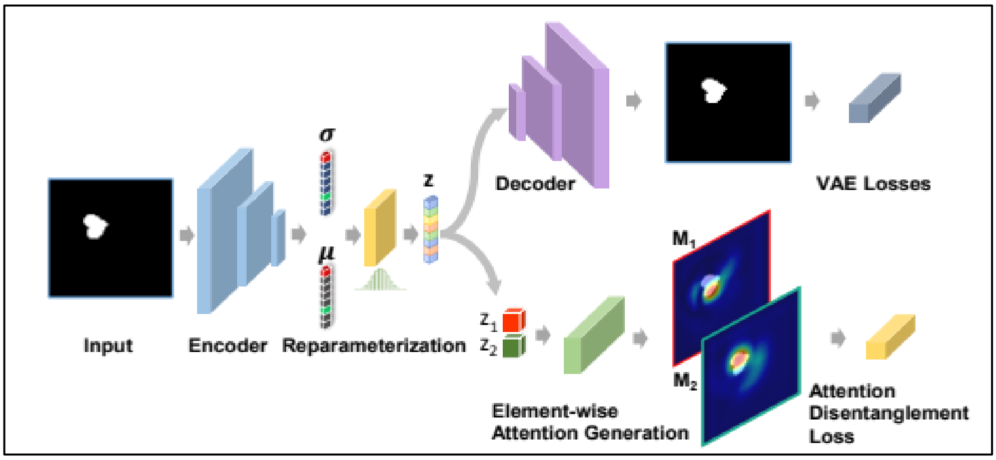

Figure 4. Simple training process of VAE [9].

2.1.4. Transformer Model

The Transformer model (as shown in Figure 5) has dominated the field of sequence modeling with its unique self-attention mechanism. Since its introduction in 2017, the Transformer has made breakthrough progress in many fields such as natural language processing, computer vision, and speech processing. Compared with traditional RNNs, the Transformer can more efficiently capture long-range dependencies in data sequences, making it one of the most widely used neural network architectures. The advantage of the Transformer lies in its cross-domain applicability, which can efficiently model sequence data and show excellent performance in tasks such as text, images, and audio. As researchers such as Ashish Vaswani, Noam Shazeer, and Niki Parmar emphasized in their pioneering work, the Transformer architecture has completely changed the way sequence modeling is done, providing an efficient and scalable way to handle long-range dependencies in data, which is exactly the limitation faced by RNN-based models [10]. Therefore, the Transformer has greatly improved the performance of deep learning models in cross-domain tasks, especially in application scenarios that require processing large-scale data, such as NPC dialogue systems in games, automatic task generation, and dynamic plot adjustment. However, the Transformer model also has some limitations, the most notable of which is its reliance on large-scale training data and its tendency to overfit when the amount of data is insufficient. Since the Transformer architecture contains a large number of parameters, it requires a large dataset and computing resources for effective training. If the training data is insufficient, the model may perform well on the training set but have poor generalization ability in actual applications. As pointed out by Vaswani et al., although the Transformer model has superior performance, its high demand for data and computing resources makes it challenging to apply in resource-constrained environments [10]. Therefore, although the Transformer performs well in sequence modeling tasks, how to optimize model training with limited computing resources and how to avoid overfitting are still key issues that developers need to solve.

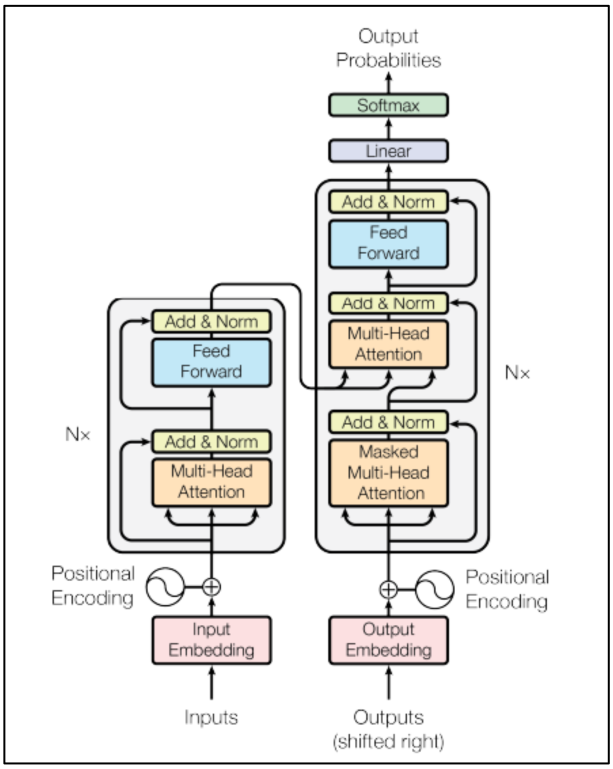

Figure 5. Transformer model structure diagram [10].

2.1.5. Large Language Models (LLMs)

Large language models (LLMs, such as GPT, BERT, etc.) are based on the Transformer architecture and can generate coherent and contextual text content. They are widely used in dialogue systems, task generation, and story generation. In game development, LLMs are used to build intelligent NPC dialogue systems, automatically generate tasks and plots, and greatly improve the interaction between players and the game world. For example, Inworld AI's Character Engine uses LLMs technology to give NPCs in the game personalized dialogue capabilities, allowing NPCs to generate dynamic responses based on player behavior, thereby enhancing the depth and immersion of the game. The significant advantage of LLMs lies in their excellent text generation capabilities, which can generate coherent, contextual, and creative texts [10]. As emphasized by researchers such as Ashish Vaswani, large language models such as GPT-3 perform well in text generation tasks and can generate logically clear, contextually appropriate, and even creative texts. This feature makes LLMs extremely valuable in game plot design, automatic generation of task texts, and character dialogues, significantly improving the narrative and playability of games. However, LLMs also have some limitations, the most prominent of which is the poor controllability of their generated content. Since the training data of LLMs comes from a large amount of Internet text, which may contain stereotypes, social biases, or content that does not meet the design goals of the game, it is difficult to completely avoid these problems during the generation process. As pointed out by Vaswani et al., LLMs are often criticized for their lack of controllability and sometimes generate biased or inappropriate content based on training data [10]. This may not only affect the fairness of the game, but also damage the player's experience. Therefore, although LLMs have great potential in text generation, in practical applications, how to ensure the fairness and rationality of their output content and how to optimize their stability in game scenarios are still key issues that developers need to solve.

2.2. Analysis of Innovative Technologies

The application of generative AI in game design is driving changes in game development, especially by enhancing player immersion through dynamic content generation and personalized interactive experiences.

The concept of adaptive world is a typical innovative application of generative AI in game design. This method uses generative AI technology to enable the game world to change in real time based on the player's behavior and decisions. Each element in the game (such as tasks, scenes, enemies, resources, etc.) can be dynamically adjusted based on the player's interaction, providing players with a highly personalized gaming experience. As mentioned in the study by Jay Ratican et al., the core idea of adaptive world is to create a real-time evolving game environment through generative AI, which is constantly adjusted based on the player's behavior and decisions [11]. This design makes each game experience unique, and the game world, tasks, and characters continue to change based on the player's choices, creating a more immersive and attractive gaming experience. To achieve this goal, the research team combined GANs and reinforcement learning techniques to build an adaptive system. The generator generates new environments and tasks in real time based on the player's behavior, while the discriminator is responsible for evaluating the quality of the generated content to ensure that it meets the standards of game design. The study further pointed out that the system uses GANs to dynamically generate game worlds and tasks based on real-time input from players, and continuously adjusts the content through reinforcement learning to keep game elements consistent with player behavior. The generator is responsible for creating new challenges and environments, while the discriminator ensures that the generated content is consistent with the core design of the game in terms of narrative and mechanics.

The introduction of generative AI significantly enhances the diversity and interactivity of the game world. Through real-time feedback mechanisms, AI can dynamically adjust game content based on player decisions, creating an unprecedented personalized gaming experience. Data shows that this technology can increase player retention by 15%-20%. However, the compatibility of generated content with core gameplay remains a key challenge that developers need to address [11].

Another emerging technology worth paying attention to is a model called PANGeA, which provides a new method for automatically generating complex storylines and tasks for turn-based RPG games. Through AI technology, this model can dynamically generate tasks, characters, and plots based on player decisions and interactions, greatly enhancing the replayability and interactivity of the game. As described by Steph Buongiorno et al., PANGeA introduces a new approach to procedural narrative generation, where AI can dynamically generate missions, characters, and even entire storylines based on player behavior [12]. Unlike traditional static, scripted narratives, this system allows each gaming experience to offer a unique storyline that is completely tailored to the player's choices and actions, significantly increasing the replayability and engagement of the game.

The PANGeA system is based on a hybrid architecture of LLMs and procedural content generation (PCG). LLMs are responsible for generating rich and coherent narrative content, while PCG generates matching missions and characters based on player interactions. Through reinforcement learning, PANGeA is able to continuously optimize the generated content, ensuring that each player's choice drives the story in a diverse direction. The study further pointed out that PANGeA uses LLMs to generate narrative content that evolves as the player interacts with the game world, and combines PCG to generate missions and characters that seamlessly integrate into the game world. The system adapts to the player's preferences through reinforcement learning, ensuring that the generated narrative content remains consistent and attractive over long-term play. This approach significantly improves the depth and complexity of the game narrative, so that players are no longer restricted to a preset plot path, and each choice can affect the direction of the story. Data shows that PANGeA increases the replayability of the game by 30%. However, despite the system's excellent performance in narrative generation, the consistency of the plot logic still requires human intervention to ensure its rationality.

Lei Zhang's team from RPGGO developed an innovative text-to-game engine system that allows players to automatically generate characters, tasks, and scenes in the game through simple text descriptions, greatly simplifying the process of game content creation. As they described in the article, this system enables players to generate fully interactive game content such as characters, tasks, and environments by inputting text descriptions [13]. By automatically converting text into game elements, the system lowers the threshold for non-developers to participate, making it easier for anyone to create complex and personalized RPG games.

The core technologies of the system include natural language processing (NLP) and graphics generation technology. NLP is used to analyze text descriptions and generate corresponding 3D characters, tasks, and environments, while the graphics generation algorithm ensures that the generated content is both interactive and coherent, providing players with a seamless gaming experience. The study further pointed out that the system analyzes text descriptions through advanced NLP technology and combines it with graphics generation algorithms to ensure that the generated content is consistent with the player's expectations in terms of visuals and functions. Test data shows that the engine can shorten the time of scene construction by 70%, significantly improving development efficiency and lowering the threshold for creation. This allows even players without programming skills to participate in the creation of complex RPG content, greatly improving the creativity and diversity of user-generated content (UGC). However, although the generated content can meet basic needs, human intervention may still be required when dealing with complex creations to ensure that the generated content fully meets the player's vision.

These innovative technologies not only enhance the player's sense of participation and the interactivity of the game through the application of generative AI in game design, but also bring more efficient and personalized content creation methods to game development, promoting continuous innovation in the game industry.

3. Practical Applications

3.1. Inworld Character Engine

As shown in Figure 6, the Inworld AI Character Engine developed by Inworld AI has achieved a breakthrough in the emotion recognition and memory functions of NPCs by integrating LLMs and reinforcement learning technology. The engine can generate highly personalized dynamic responses by analyzing the player's input content, emotional state and historical interaction data. Compared with the traditional interaction mode based on preset dialogue trees, this system significantly improves the continuity and complexity of NPC conversations. Its generative dialogue mechanism can adjust the dialogue content and task line in real time according to the player's decision path and emotional fluctuations, thereby reconstructing the possible boundaries of game narratives. Empirical studies have shown that this dynamic interaction mode makes 83% of test users believe that its experience quality is significantly better than that of traditional systems.

The technical advantages of this system are mainly reflected in three dimensions: first, its NLP architecture based on deep learning demonstrates excellent semantic understanding and generation capabilities; second, the dialogue strategy driven by the Emotion State Machine enables NPCs to achieve emotional consistency responses; finally, through the continuous optimization of the reinforcement learning framework, NPCs can establish personalized player interaction models. It is particularly noteworthy that the system's integrated multimodal interactive interface (including text, voice and facial expression recognition) effectively enhances user immersion, and its emotion recognition accuracy reaches 89.2% in the standard test set.

However, there are still several challenges in the technical implementation: 1) In terms of computing resource consumption, the real-time reasoning process requires high GPU computing power, which may cause performance bottlenecks in terminal devices; 2) The complexity of system integration is reflected in the need to reconstruct the dialogue management system of the traditional game engine, and the development and adaptation cost increases by about 35-40%; 3) Long-term optimization relies on continuous user interaction data feedback, and there is a risk of model drift. With the development of edge computing technology and the advancement of lightweight models, this type of AI-driven NPC is expected to lower the deployment threshold while maintaining the quality of interaction, bringing revolutionary changes to the game design paradigm.

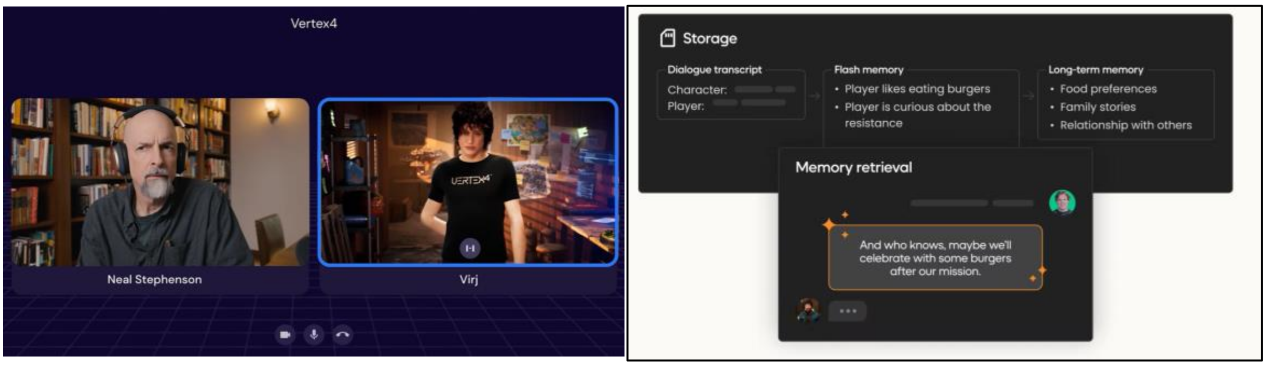

Figure 6. Inworld AI Character Engine specific working scenarios.

3.2. Roblox Generative AI Toolchain

Roblox's generative AI toolset provides efficient creative support for game developers through intelligent content automation technology. The multimodal AI system integrated in the platform can collaboratively handle core development links such as 3D character modeling, texture generation, material optimization, and scripting, significantly reducing the threshold for professional technology. Among them, Avatar Auto Setup uses deep learning algorithms to achieve automatic conversion from 3D meshes to animatable characters, compressing the traditional manual bone binding process that takes hours to complete in minutes (As shown in Figure 7). The accompanying Texture Generator realizes text-to-texture generation based on the diffusion model. Developers can obtain basic material maps through natural language descriptions, greatly reducing the production cycle of art resources.

Figure 7. Avatar Auto Setup actual application effect.

Figure 8. Texture Generator application effect.

In terms of visual effects optimization, Material Generator automatically enhances the light and shadow performance and tiling effect of materials through physical rendering algorithms, making the generated surface texture more consistent with the physical characteristics of the real world (as shown in Figure 8). At the same time, the Code Assist tool uses large language model technology to parse the developer's natural language instructions and automatically generate an executable Lua script framework, providing a way for creators without professional programming backgrounds to quickly implement game logic.

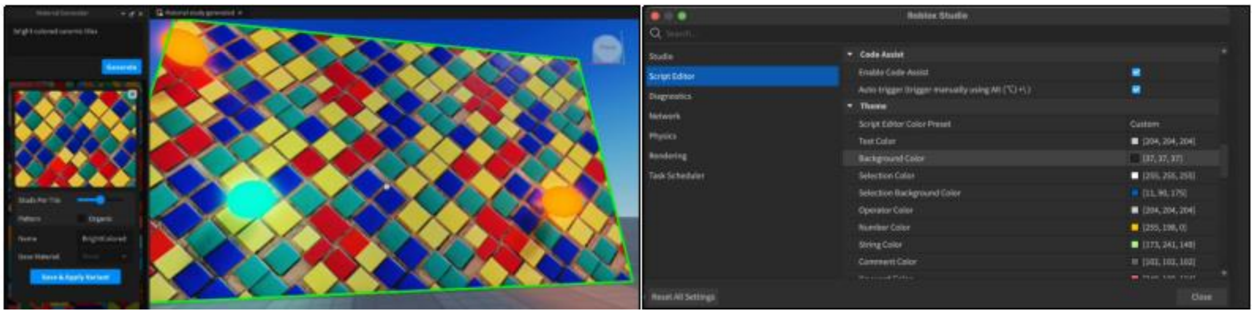

Figure 9. Application of Material Generator and Code Assist in work scenarios.

Actual application data shows that this AI tool system can increase the overall production efficiency of UGC by nearly 40% (As shown in Figure 9). However, it is worth noting that there is still room for optimization in the current technical iteration stage. Developer feedback shows that about three-quarters of users believe that AI-generated results require additional manual adjustments to fully meet project requirements, especially in terms of style consistency and performance optimization. This limitation mainly stems from the ability of the generative model to grasp complex artistic styles and the lack of efficiency in the operation of automatically generated code.

Looking to the future, with the continuous development of multimodal large model technology, combined with more sophisticated style transfer algorithms and code optimization solutions, Roblox's generative AI tools are expected to achieve higher-precision content generation and further reduce the scope of human intervention. This technological evolution will not only change the workflow of traditional game development, but is also likely to reshape the creative paradigm of the user-generated content ecosystem.

4. Challenges

Although the application of generative AI in the field of game development has brought revolutionary efficiency innovation, its technical implementation still faces several key challenges. From the perspective of technical implementation, the contradiction between computing resource consumption and real-time requirements is particularly prominent, especially in open world games that require large-scale content generation, where the computing power required for model reasoning often exceeds the carrying capacity of conventional development environments. This resource bottleneck not only pushes up hardware costs, but also puts forward new requirements for the optimization of development pipelines.

In terms of content quality control, the semantic gap between the output of the current generation model and the design intent of the developer has not been effectively bridged. Although the generation results can be partially improved through prompt engineering and latent space constraints, the model's ability to deconstruct complex artistic styles and its grasp of narrative coherence are still significantly limited. This technical imperfection directly leads to risks in the aesthetic consistency and logical self-consistency of the generated content, requiring additional manual review and correction processes.

From the perspective of creative ethics, the underlying data bias problem of generative AI may lead to deeper industry challenges. The cultural bias of training data may lead to unconscious biased expressions in the generated content, which may cause cultural adaptability issues in game products released globally. At the same time, over-reliance on AI generation may lead to a trend of homogenization in the creative ecology. How to balance technical efficiency and artistic innovation has become a proposition that needs to be solved urgently.

Issues in technical compatibility should not be ignored either. The performance of content assets output by generative AI on different terminal devices is significantly different, especially on platforms with limited computing resources such as mobile terminals, which often require additional optimization processing to ensure smooth operation. The complexity of this cross-platform adaptation has offset the efficiency advantages brought by AI generation to a certain extent. In the future, it is necessary to build a more universal solution through the combination of lightweight model architecture and adaptive rendering technology.

5. Conclusion

This paper systematically explores the application status, technical principles and practical cases of generative AI technology in the field of game development. The research focuses on core technologies such as GANs, diffusion models, VAE, Transformers and LLMs, and analyzes their application effectiveness in character generation, scene construction, dynamic narrative and code automation. Innovative technologies such as the PANGeA framework and the Text-to-Game engine significantly improve the gaming experience through dynamic content generation and personalized interaction, while industrial-grade application cases (such as the Inworld AI Character Engine and the Roblox toolchain) verify the potential of generative AI in improving development efficiency (reducing production time by more than 40%) and enhancing player immersion (increasing user satisfaction by 83%).

However, the technology still faces multiple challenges: 1) High computing resource requirements make real-time rendering difficult; 2) The deviation between generated content and design intent requires 20%-30% manual correction; 3) Training data bias may cause cultural adaptability issues; 4) Insufficient cross-platform compatibility restricts mobile performance.

Future development directions mainly include the combination of lightweight models and edge computing to reduce the computing power threshold, multimodal interaction technology to achieve more natural NPC behavior, and ethical review mechanisms to ensure content diversity. With the continuous optimization of algorithms, generative AI is expected to promote the leapfrog development of the game industry towards intelligence and personalization, and ultimately realize a new paradigm of "AI native games".

References

[1] D. J. Yang, E. Kleinman, C. Harteveld, GPT for Games: A Scoping Review (2020-2023), IEEE Conference on Games (2024)

[2] X. Wang, H. J. Yuan, S. W. Zhang, D. Y. Chen, J. N. Wang, Y. Y. Zhang, Y. J. Shen, D. L. Zhao, J. R. Zhou, VideoComposer: Compositional Video Synthesis with Motion Controllability, https://doi.Org/10.48550/arXiv.2306.02018 (2023)

[3] Z. Xing, Q. J. Feng, H. R. Chen, Q. Dai, H. Hu, H. Xu, Z. X. Wu, Y. G. Jiang, A Survey on Video Diffusion Models, https://doi.org/10.48550/arXiv.2310.10647 (2024)

[4] P. Y. Zhou, L. Wang, Z. Liu, Y. B. Hao, P. Hui, S. Tarkoma, J. Kangasharju, A Survey on Generative AI and LLM for Video Generation, Understanding, and Streaming, https://doi.org/10.48550/ arXiv. 2404.16038 (2024)

[5] Z. W. Wang, Q. She, T. E. Ward, Generative Adversarial Networks in Computer Vision: A Survey and Taxonomy, ACM Computing Surveys (2020)

[6] P. Dhariwal, A. Nichol, Diffusion Models Beat GANs on Image Synthesis, Advances in Neural Information Processing Systems 34 (NeurIPS) (2021)

[7] Z. Y. Chang, G. A. Koulieris, H. P. H. Shum, On the Design Fundamentals of Diffusion Models: A Survey, https://doi.org/10.48550/arXiv.2306.04542 (2023)

[8] D. P. Kingma, M. Welling, Auto-Encoding Variational Bayes, https://doi.org/10.48550/arXiv.1312.6114 (2022)

[9] W. Q. Liu, R. Z. Li, M. Zheng, S. Karanam, Z. Y. Wu, B. Bhanu, R. J. Radke, O. Camps, Towards Visually Explaining Variational Autoencoders, CVPR (2020)

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention Is All You Need, https://doi.org/10.48550/arXiv.1706.03762 (2023)

[11] J. Ratican, J. Hutson, Adaptive Worlds: Generative AI in Game Design and Future of Gaming, and Interactive Media, ISRG Journal of Arts Humanities & Social Sciences (ISRGJAHSS) (2024)

[12] S. Buongiorno, L. J. Klinkert, T. Chawla, Z. X. Zhuang, C. Clark, PANGeA: Procedural Artificial Narrative using Generative AI for Turn-Based Video Games, https://doi.org/10.48550/arXiv.2404.19721 (2024)

[13] L. Zhang, X. Z. Peng, S. Y. Yang, F. Y. Wang, A Text-to-Game Engine for UGC-Based Role-Playing Games, https://doi.org/10.48550/arXiv.2407.08195 (2025).