Proceedings of the 4th International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/2026.34137

# *Application and Analysis of AIGC Dynamic Generation Technology in Open World Games*

**Qiuyu Ruan**

*School of Artificial Intelligence, Hainan Normal University, Haikou, China*
*202324120242@hainnu.edu.cn*

***Abstract.*** Open World Games have a high degree of freedom exploration, non-linear narration and constantly changing world state, which puts forward higher system requirements for dynamic content generation than linear games. This paper analyzes the application of AIGC in open world games, focusing on three main technical paths: text generation for NPC and narrative, multimodal generation for environment and asset production, and programmed content generation based on machine learning. This paper further summarizes the key supporting conditions of these technologies from the four aspects of knowledge enhancement, long-term memory, design controllability and runtime deployment. Research shows that AIGC in the open world not only improves the efficiency of content generation, but also maintains a stable balance among consistency, real-time and controllability. Among them, text generation relies on Transformer and retrieval enhancement mechanism to maintain the consistency between roles and world knowledge, long-term memory mechanism is used to support the continuity of roles in the process of continuous interaction, multimodal generation is mainly responsible for the production of concept maps and asset prototypes, and the key to the generation of programmed content is whether the results can meet the play structure and system constraints. Overall, the core challenge of the open world AIGC is not whether to generate content, but whether the generated results can be truly connected to the game system at runtime and form a stable closed loop between the world state, player interaction and design control.

***Keywords:*** AIGC, Open world games, NPC dialog generation, Knowledge enhancement, Programmed content generation

## 1. Introduction

The core feature of open world games is that players can complete exploration, interaction, narrative promotion and spontaneous goal construction with fewer linear constraints. Therefore, the system should not only support large-scale space, enrich roles and multi-line tasks, but also provide reliable feedback when players deviate from the "main path". Although the traditional script tree, fixed dialog table and static task chain have advantages in controllability, they are prone to problems such as repetition, insufficient coverage and context disconnection in the face of high free input, continuous state change and long-term interaction [1, 2]. The reason why AIGC technology is regarded as an important increment of open world games is not that it simply replaces screenwriters,

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.34137

art or level designers, but that it provides a new implementation path for dynamic generation at runtime: NPCs can generate responses according to the context, and the task text can also change with the world state [1, 2]. In addition, concept maps, environment sketches and some 3D prototypes can also be generated by text conditions [3, 4].

However, the requirements of the open world scenario for AIGC are not just to generate relevant game content, but also to determine whether the generated content can be used by the game system. Based on the existing research, the related systems are subject to at least three rigid constraints. First, world consistency requires that the generated content comply with the world background, role setting, location relationship, task preconditions and historical events. This problem usually depends on external knowledge organization and retrieval mechanisms [5, 6]. Second, real-time requires the model to complete reasoning within an acceptable delay, and its feasibility is directly affected by the model size, hardware budget and deployment scheme [7]. Third, the design controllability requires the system to maintain the author's intention, plot boundary and character stability while preserving the free input of players. This problem has gradually expanded from the prompt project to the narrative planning and verification chain [8-10]. Based on this, this paper focuses on the task types, supporting mechanisms, performance tradeoffs and system bottlenecks in the open world, and discusses the future system form.

## 2. Methods

### 2.1. Technical path

#### 2.1.1. NPC and narrative-oriented text generation

AIGC in the open world is first embodied in text generation, namely NPC dialogue, dynamic narrative and task text. The current mainstream scheme is based on transformer, and its core is self attention mechanism. Set the input sequence as X, then each token is mapped to a query, key, and value vector:

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V \tag{1}$$

Single head attention can be written as:

$$Attention\,(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

At the application level, the existing LLM-NPC system usually splices the role identity, task status, player input and historical dialogue into the same context, and the model completes the conditional generation [1, 2]. In the open world scenario, this means that each token currently generated by NPC does not only depend on the previous word, but dynamically reads the relationship between "role setting - current world state - Dialogue history - external knowledge". Therefore, the large language model can replace the fixed dialog tree to complete runtime generation, but it will also expose two direct problems: first, the role setting is easy to be diluted in the long context; Second, the model is prone to setting drift when it lacks external fact binding. For this reason, existing studies have begun to shift from static personality input to character-based

DOI: 10.54254/2755-2721/2026.34137

dialogue, and include character identity, relationship changes, emotional style and expression habits into conditional control [11]. From the perspective of the system, NPC in the open world should not be understood as a single round of text generation node, but as a role agent with persistent state and relational memory.

### 2.1.2. Retrieval enhancement generation for world knowledge consistency

In terms of knowledge consistency, RAG is currently the most commonly used middle-level solution. The basic idea is to combine the parametric generator with the non-parametric external knowledge base, retrieve it first, and then generate it. For a given input x, the retriever estimates the document conditional probability pη(z|x),then the generator estimates:

$$p\left(yx\right)=\sum_{z\in top-k}p_{\eta}\left(zx\right)p_{\theta}\left(yx,z\right) \tag{3}$$

Where z represents the retrieved world knowledge fragments, task logs or role files [5]. The significance of this mechanism is that the world outlook update does not need to retrain the model, but only needs to update the external index. For open world games, RAG is especially suitable for processing documentary knowledge such as location description, task record, camp announcement and role background; The knowledge map enhancement method is more suitable for dealing with structural relationships such as "who belongs to whom", "which task depends on which task" and "which two places are connected", because its retrieval object is not an independent paragraph, but a sub graph related to the current context [6]. Therefore, the essence of the consistency problem is not just "whether knowledge is sent into the model", but "whether knowledge enters the generation process with the correct structure".

### 2.1.3. NPC long-term memory mechanism for continuous interaction

Long term memory is another basis for the credibility of NPC in the open world. If only relying on the context window, the model finds it difficult to maintain the identity, relationship and event continuity across time periods. The memory reflection planning framework proposed by Generative Agents provides a clearer implementation idea [12]. The system first writes player interaction, task changes and world events into memory entries; Then the memory is sorted according to novelty, relevance and importance. The common form can be written as follows:

$$s=\alpha R_{c}+\beta R_{l}+\gamma I \tag{4}$$

Where Rc stands for recency, Rl for relevance, and I for importance. High score memory enters the current prompt, and high importance events are further compressed into reflective conclusions, such as "players are trustworthy" and "a camp is more sensitive to outsiders", which are then transformed into subsequent dialogue strategies and behavior preferences by the planning module. Compared with static character cards, this structure allows characters to maintain a relatively stable behavior trajectory in long-term interaction. For the open world, this means that the role will not only behave like a person living under a specific world view in the current round, but still show continuity after many encounters, task advancement and relationship changes.

DOI: 10.54254/2755-2721/2026.34137

### 2.1.4. Multimodal generation for environment and asset production

Developing in parallel with text generation is a multimodal generation path for environment and asset production. Text to image model maps natural language description to visual results by learning the joint distribution of text and text, and can be used for rapid generation of concept map and scene sketches [3]. In the 3D direction, the Text to 3D method usually does not directly predict the available game assets. Instead, it is expressed as NeRF or rough geometry, and then enters the production pipeline through topology, material and binding processing [4]. The scheme with diffusion prior as the core can be written in the form of iterative denoising:

$$x_{t-1} = f_\theta(x_t, c) \tag{5}$$

Where c is the text condition. This path is suitable for the rapid production of concept maps and prototype assets, but in the open world, the "visible" visual results do not mean that they can be directly applied to the game, because the formal assets must also meet the requirements of collision bodies, bones LOD, Performance budget and engine state machine. It can be seen that multimodal AIGC is more suitable as a prototype production tool at the current stage, rather than an automatic generator that completely replaces the semantic integrity of the engine.

## 2.2. Runtime support mechanism of open world AIGC

Programmed content generation constitutes the third key technology path. The traditional Search-based PCG relies on the explicit fitness function to constantly search for content that meets the playability, difficulty and resource constraints in the candidate space; PCGML emphasizes learning from the existing content distribution, and then generates new levels, regions or event templates with the same style [13]. If the generator is G(z), its essence is to learn the content distribution on the latent variable z, and the subsequent search selects a more appropriate sample according to the objective function F(G(z)). This idea shows that automatic content generation in the open world can not only pursue the matching degree with existing content, but also meet the constraints of game structure, closed-loop state machine and resource allocation. Because of this, the key to programmed content generation in the open world is not only the degree of automation, but also the compatibility between the generated results and existing systems.

Based on the above task pedigree, the key to the entry of open world AIGC into the runtime system is not the single model capability, but the middle-level support mechanism. The first is knowledge enhancement, which is responsible for replacing static training knowledge with renewable world knowledge; The second is long-term memory, which is responsible for promoting one-time interaction to a sustainable role state; Thirdly, the controllability of design, which embeds the author's intention stably into the open generation through structured prompt, narrative intermediate representation and verification system; The last is the deployment mechanism, which reduces the irrelevant context through vector retrieval, absorbs the worldview corpus with lightweight model and parameters, and then finds a compromise between delay, quality and cost through quantification and cloud edge mixing [7, 14]. The resulting system evaluation dimensions can be summarized as consistency, real-time and controllability. Without consistency, the higher the player's degree of freedom, the more problems the system exposes; Without real-time, even high-quality generation will destroy the sense of immersion; Without controllability, the system will continuously deviate from the author's intention and task structure under open input. The three are

DOI: 10.54254/2755-2721/2026.34137

not independent of each other, but restrict each other: consistency often increases the cost of retrieval and verification, real-time requires compression of context and model scale, and controllability requires the introduction of additional rule layers and verification layer.

### 2.3. Main challenges of the open world AIGC

The resulting systemic bottlenecks are mainly reflected in four aspects. First, there is still a gap between multimodal generation and game engine semantics. The current model optimizes the visual quality, rather than colliding body, bone, material specifications and status updates. Second, text generation has not been stably connected to the task state machine, environment simulation and world update mechanism, and many systems still remain in the interpretation of the world view, rather than reliably changing the state of the world. Third, the conflict between long-term memory and external knowledge is still immature, and it is prone to inconsistency between personal experience, dialogue history and world setting. Fourth, the man-machine collaboration process has not been standardized. Although generative AI improves the efficiency of conceptual design and prototype iteration, it also brings problems such as the weakening of authorship, style convergence and the rising cost of manual trimming. In this regard, the core challenge of open world AIGC is no longer whether to generate corresponding content, but how to form a stable closed loop between runtime generation, system constraints and design control.

## 3. Future direction

The future open world AIGC system is more likely to present a hierarchical collaborative architecture. At the bottom, vector retrieval, knowledge mapping and world state database are used to ensure the consistency of facts; In the middle layer, memory, reflection, planning and verification modules ensure the continuity of roles and narrative boundaries; In the upper layer, the designers provide abstract narrative goals, aesthetic constraints and style rules, so as to transform the free generation mode into the generation mode within the constraint boundary. In this process, NPC will gradually evolve from a dialogue generation node to a generative agent with the ability of perception, memory and action, and the open world will also shift from the traditional paradigm of static world and local dynamic text to the content ecology of sustainable evolution.

At the same time, the research also needs to further solve three problems: firstly, how to establish a multimodal semantic interface for the game engine, so that the AIGC results naturally contain animation, collision, performance budget and status update information; Secondly, how to build a quantifiable long-term evaluation framework to measure role consistency, narrative convergence, player participation and system fairness; Thirdly, how to form a clearer collaboration specification at the industrial level, so that generative AI can become an amplifier for designers instead of a black box system controlled by authors.

## 4. Conclusion

In this paper, the AIGC dynamic generation technology in open world games is structurally analyzed. The focus of this article is not only the ability comparison of different generation models, but also how these technologies enter the system to run and assume the actual functions in the open world scenario. Based on the text analysis, the current research can be summarized into three main technical paths: first, text generation for NPC dialogue, task text and dynamic narrative; The second is multimodal generation for concept maps, scenario sketches and asset prototypes; The third is

DOI: 10.54254/2755-2721/2026.34137

procedural content generation for regions, checkpoints and event templates. On this basis, knowledge enhancement, long-term memory, design controllability and runtime deployment constitute the common support mechanism for the implementation of these technologies. A black box system controlled by the operator.

It can be seen from the full text that AIGC in the open world has obviously expanded the capability boundary of the game system in terms of role interaction, content expansion and prototype production, but the key problem is not whether the generated content meets the technical requirements, but whether the generated content is consistent with the world knowledge, role status, task structure and system constraints. Text generation improves the naturalness of interaction, but it faces the problems of setting drift and role continuity; Retrieval enhancement and long-term memory can help alleviate this problem, but at the same time, it increases the system complexity and runtime overhead; Multimodal generation improves the asset production efficiency, but it is still difficult to directly replace the available assets of the engine; Programmed content generation improves the ability of content expansion, but the real difficulty is whether the generated results can meet the closed-loop constraints of play and state machine. It can be seen that the core challenge of the open world AIGC has shifted from "whether the model can be generated" to "whether the system can stably accept the generated results".

Therefore, the follow-up research should not only focus on the larger model or stronger single generation ability, but should continue to promote the collaborative development between knowledge organization, long-term memory, narrative control, multimodal engine interface and deployment optimization. Only when the generated results can meet the requirements of world consistency, real-time interaction and design controllability at the same time, can AIGC further evolve from a local enhancement tool in the open world to a system capability that supports the continuous operation of the content ecosystem.

## References

[1] Albino, P. M. S. (2025). Integrating GPT-based language models into NPCs for realistic dialogue and contextual awareness in unity games.

[2] Mulyana, A., Wibisono, Y., & Anisyah, A. (2025). Non-playable characters based on large language models for role playing games (RPG). Brilliance: Research of Artificial Intelligence, 5(2), 785–792.

[3] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., et al. (2021). Zero-shot text-to-image generation. In Proceedings of the International Conference on Machine Learning (pp. 8821–8831). PMLR.

[4] Poole, B., Jain, A., Barron, J. T., & Mildenhall, B. (2022). Dreamfusion: Text-to-3d using 2d diffusion. arXiv. https: //doi.org/10.48550/arXiv.2209.14988

[5] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459–9474.

[6] Kang, M., Kwak, J. M., Baek, J., & Hwang, S. J. (2023). Knowledge graph-augmented language models for knowledge-grounded dialogue generation. arXiv. https: //doi.org/10.48550/arXiv.2305.18846

[7] McGrath, J., Lorandi, M., & Belz, A. (2025). Echoes of others: Real-time LLM dialogue generation for immersive NPC interaction. In Proceedings of the 18th International Natural Language Generation Conference: System Demonstrations (pp. 1–2).

[8] Dong, H. (2025). A possible prompt structure for AI director in narrative games.

[9] Wang, Y., Zhou, Q., & Ledo, D. (2024). StoryVerse: Towards co-authoring dynamic plot with LLM-based character simulation via narrative planning. In Proceedings of the 19th International Conference on the Foundations of Digital Games (pp. 1–4).

[10] Buongiorno, S., Klinkert, L., Zhuang, Z., Chawla, T., & Clark, C. (2024). PANGeA: Procedural artificial narrative using generative AI for turn-based, role-playing video games. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 20(1), 156–166.

[11] Zhou, J., Chen, Z., Wan, D., Wen, B., Song, Y., Yu, J., et al. (2023). CharacterGLM: Customizing chinese conversational AI characters with large language models. arXiv. https: //doi.org/10.48550/arXiv.2311.16832

DOI: 10.54254/2755-2721/2026.34137

[12] Park, J. S., O'Brien, J., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. In Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (pp. 1–22).
[13] Summerville, A., Snodgrass, S., Guzdial, M., Holmgård, C., Hoover, A. K., Isaksen, A., et al. (2018). Procedural content generation via machine learning (PCGML). IEEE Transactions on Games, 10(3), 257–270.
[14] Nakazawa, Y., & Sakai, M. (2025). Examination of model efficiency and its application in the game field for Japanese large-scale language models. In Proceedings of the International Conference on Asia Digital Art and Design Asia Digital Art and Design Association (pp. CT3101–CT3107). Asia Digital Art and Design Association.