Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 12, No. 4, 2024, 583-597.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Review paper

Harnessing Machine Learning for Procedural Content Generation in Gaming: A Comprehensive Review

Shaghayegh Saffari, Morteza Dorrigiv* and Farzin Yaghmaee

Electrical and Computer Engineering Department, Semnan University, Semnan, Iran.

Article Info

Article History:
Received 15 September 2024
Revised 08 December 2024
Accepted 08 January 2025

DOI:10.22044/jadm.2025.15016.2603

Keywords:
Procedural Content Generation, Machine Learning, Artificial Intelligence.

**Corresponding author: dorrigiv@semnan.ac.ir (M. Dorrigiv).*

Abstract

Procedural Content Generation (PCG) through automated and algorithmic content generation is an active research field in the gaming industry. Recently, Machine Learning (ML) approaches have played a pivotal role in advancing this area. While recent studies have primarily focused on examining one or a few specific approaches in PCG, this paper provides a more comprehensive perspective by exploring a wider range of approaches, their applications, advantages, and disadvantages. Furthermore, the current challenges and potential future trends in this field are discussed. Although this paper does not aim to provide an exhaustive review of all existing research due to the rapid and expansive growth of this domain, it is based on the analysis of selected articles published between 2020 and 2024.

1. Introduction

Procedural Content Generation (PCG) refers to the automatic generation of game content algorithmically. It is one of the most active topics within both the gaming industry and academic studies. One of the primary expenses in video game development is content creation. Game development generally requires assembling a team possessing diverse skills, entails substantial time commitments, and demands significant budget allocations, which culminate in a limited quantity of game content [1]. Procedural generation can reduce the need for resources typically produced manually, requiring minimal or even no human input [2].

The gaming industry has rapidly grown into a multi-billion-dollar sector, with over 37 million active players on Steam daily [3]. To meet the rising demand for engaging, high-quality content, research in game development and design is crucial. PCG addresses these challenges by automating content creation during both development and gameplay [1].

PCG is used for various design goals, initially aiming to improve data compression, as seen in games like Elite, where procedural galaxies allowed vast exploration. Rogue-like games use PCG to control generated content while enhancing replayability and adaptability. Recent advancements focus on adapting games to player preferences [4].

Research indicates there is no single best method for content generation; instead, multiple approaches are used depending on the content type. Further research is needed to evaluate PCG methodologies and explore broader applications in game development [5].

Artificial intelligence in PCG enhances replay value, reduces production costs and effort, and enables autonomous generation and data compression. Additionally, PCG is well-suited for co-creativity, mixed-initiative design, content repair, critique, and analysis. The structure of this content can differ greatly, and various techniques, including machine learning, can be employed for its generation [6]. Many Machine Learning (ML) methods can be utilized in a generative role. These include autoencoders [7], adversarial learning [8], Reinforcement Learning (RL) [9], transfer learning [10], deep learning [8], Diffusion Models [9], Transformers [11], and others [6]. The basic idea is to train a model on instances sampled from some distribution, and then use this model to produce new samples [9].

A substantial body of research has been conducted in the field of PCG. Additionally, review articles often emphasize categorization based on the types of content and the methods used to generate them [5,12,13], or they focus on broad categories such as PCG through ML [14,15]. However, there is less focus on a comprehensive classification based on ML methods and the examination of their specific applications in procedural content generation. In this regard, only a few articles exist [6,8], and more than three years have passed since their publication. Consequently, given the rapid advancement of ML models, these articles lack coverage of newer, more widely-used models that have improved PCG in games. Moreover, recent articles tend to explore the role of only specific categories of ML methods, such as adversarial approaches or Large Language Models (LLMs), in game content generation [11,16]. As a result, recent research does not provide a comprehensive classification based on ML methods nor an analysis of their specific applications and challenges in procedural content generation, particularly considering the newer approaches discussed in recent literature.

Therefore, this paper proposes a new taxonomy to categorize methodologies in machine learning, specifically for PCG. This taxonomy draws on current trends in research that predominantly employ specialized ML technique through PCG. Furthermore, previous review studies [5,6,8,11-15] have highlighted a limited number of significant challenges in this area, providing a general overview. However, they do not extensively discuss the important challenges of PCG in machine learning.

This paper thoroughly examines the main challenges of PCG through ML. These challenges arise from the fundamental differences between game development and other ML domains. Notably, game development datasets are usually smaller, represent dynamic systems dependent on user interaction, and often lack readily available high-quality data. Despite these challenges, they can be effectively managed with the right ML techniques.

In this study, we explore the ML methods that enhance PCG's effectiveness in games. We discuss the challenges encountered and provide detailed examples of algorithmic applications.

The rest of this paper is organized as follows: Section 2 introduces types of game content. The types of ML methods used in game content generation are presented in Section 3. In addition, Section 4 discusses and presents the open problems of PCG using ML methods. Finally, Section 5 concludes the paper.

2. Content Types

Most game content needs to be coherent to maintain aesthetic appeal. Key content types, like level design in most games and storyline creation in narrative-driven games, are essential for a positive player experience, where the primary goal is playability and coherence. Other content, such as minor character sprites or textures for less significant scenes, can be imperfect for greater efficiency [9]. Game content includes various elements like levels, items, and characters [11].

Hendrikx et al. [17] proposed a taxonomy categorizing game content into six levels: bits, space, design, scenarios, systems, and derivatives. Game bits are the basic units like textures, sounds, music, and visual effects, with some elements, such as character textures, being essential, while others like sound effects are optional. Game spaces refer to the environments where the game occurs, including maps and terrain. Game design encompasses the rules and mechanics, while scenarios organize levels into cohesive sequences of events, such as puzzles and stories. Complex environments are often represented by systems like ecosystems or urban networks, and derivative content includes additional elements like leaderboards and news items that enhance the overall game experience.

3. Machine Learning

Artificial Intelligence (AI) is a key branch of computer science that enables human-like behavior across a wide range of tasks, from simple to complex ones. As shown in Figure 1, ML, a significant subset of AI, empowers systems to learn from data without explicit programming [18].

ML techniques can be categorized into various related architectures and applications. While researchers have proposed different classifications for ML-based approaches [6,9,18], this article adopts a hierarchical structure, as shown in Figure 1, to provide a more comprehensive perspective by categorizing some of the most popular ML techniques used in PCG for games. Additionally, in this section and its subsections, the applications, advantages, and challenges of these approaches in PCG for games are gradually examined, along with the rationale behind the positioning of each approach within the hierarchical structure.

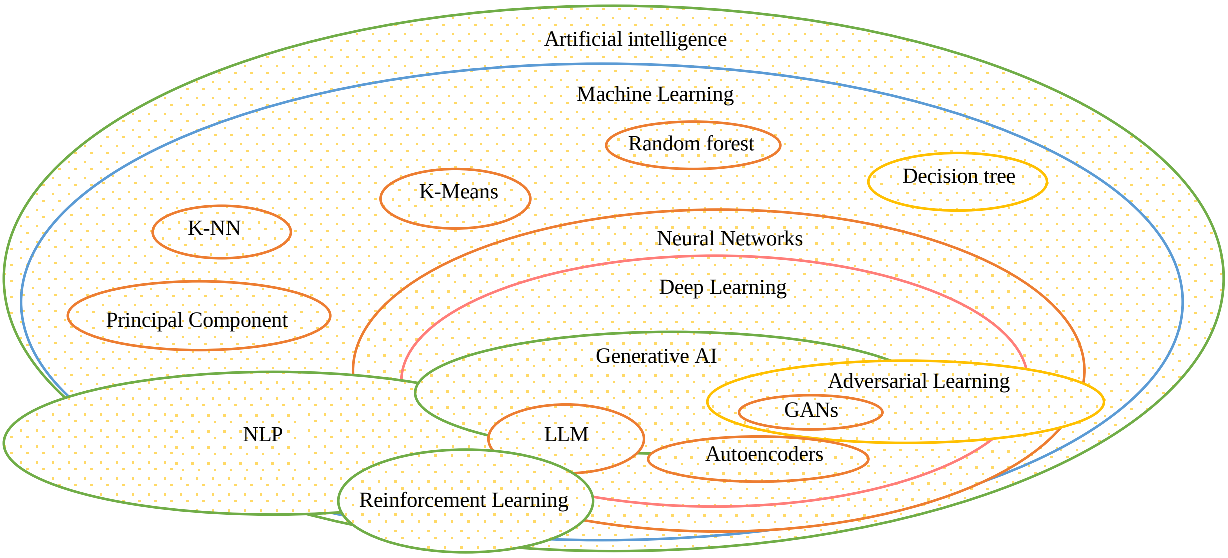

Figure 1. The hierarchy among machine learning method.

These techniques, widely utilized in PCG for games, are divided into four main categories based on their learning paradigms: supervised learning, unsupervised learning, semi-supervised learning, and RL [8]. Supervised learning, used in regression and classification tasks, relies on labeled data and models like Support Vector Machines and Decision Trees [6]. These models play a significant role in predicting outcomes such as win/loss in games, matching challenges to user skills, and assessing content quality [14]. Unsupervised learning, using techniques like clustering (e.g., K-means) and Principal Component Analysis (PCA), identifies patterns in unlabeled data [9]. Additionally, models such as autoencoders are applied to content generation, like designing game levels [8]. Semi-supervised learning, which combines labeled and unlabeled data, is used in scenarios where labeling is costly [6]. Reinforcement learning, effective in sequential decision-making, adapts dynamically to complex environments, making it highly impactful in robotics and games [9].

Neural networks, inspired by the human brain, are the main tools in ML, capable of identifying complex patterns. They are used for generating content like game levels and 3D objects [19]. In contrast, as illustrated in Figure 1, ML also includes classical techniques that are not dependent on neural networks. Classical ML techniques include interpretable and powerful models like K-Nearest Neighbors (KNN), Decision Trees, Random Forests, and PCA [18]. These methods find applications in tasks such as analysis and prediction during the PCG process in games [9].

Decision Trees, while simple and efficient, can be prone to overfitting. Random Forests, as an advanced version, are more robust and suitable for complex datasets, albeit requiring more computational resources [20]. Additionally, the KNN algorithm classifies data based on the nearest neighbors and is well-suited for quick predictions in small datasets [21]. For instance, Stollas et al. [21] developed Florescence, a rhythm game that adapts to player preferences. This game analyzes users' music tracks using Librosa, predicts genres using algorithms like Random Forests and KNN, and customizes the gaming experience based on the user's skill level. In another example, the Learning-Embedded Attribute Petri Net model optimizes learning in serious games by simulating gameplay and decision-making. It combines RL for guiding behaviors and uses Random Forest classification for decision support through decision tree analysis [20]. Also, PCA aids in reducing data dimensions while retaining key features, though there is a risk of losing important information. Moreover, K-means, applied in clustering, is useful for data analysis but its accuracy depends on the selection of the number of clusters. Consequently, K-means and PCA have been employed for data analysis and dimensionality reduction to generate visuals in the online game Crea.Blender SDG [22]. Additionally, Table 1 provides a concise summary of the applications, advantages, and disadvantages of this studies mentioned in this section.

Table 1. An overview of the applications of classic ML techniques in PCG.

Model/Algorithm	Application	Advantages	Disadvantages/challenges	References
Random Forests	Classification of player performance to improve player learning in a serious game	Real-time	Computational complexity	[20]
KNN and Random Forests	Genre prediction of songs for adapted Rhythm generation to the user in Efflorescence game	Personalized	Need for more data	[21]
K-means and PCA	Data analysis and data dimensionality reduction for image generation in online game crea.blender SDG	Real-time, Interactive	Bias in human evaluators	[22]

In conclusion, classical ML techniques, due to their interpretability and ability to address problems with well-defined features and structured data, play a vital role in tasks involving less complexity. While not directly applied for content generation, they are instrumental in fast predictions and classifications within the PCG processes.

3.1. Deep Learning

Deep learning (as shown in Figure 1) is a subset of machine learning, defined as neural networks with at least two layers and some degree of nonlinearity [8]. These methods have achieved significant success in PCG for games.

For instance, one of the most widely used deep learning models in PCG is convolutional neural networks (CNNs). These models have been successful in generating game images and dynamic characters. For example, CNNs have been employed in Final Fantasy XV and The Sims games to create characters based on facial images [23]. Additionally, Stephens et al. [24] utilized CNNs to balance levels and predict player wins in the shooter Nuclear Throne game.

The main advantage of CNNs lies in their ability to extract complex visual features. However, these approaches are limited when processing sequential data. In contrast, Long Short-Term Memory (LSTM) networks have demonstrated high performance in generating game levels using sequential data, such as path sequences and player movements in the Lode Runner game [25]. Moreover, Inns et al. [26] used bidirectional LSTMs and fuzzy-GA models to procedurally generate game levels. While these models preserve long-term memory, they are constrained by computational resources [26].

WaveFlow is another model that has been used to improve the quality and diversity of sound effects in games. This model provides a richer gaming experience by enabling controllable sound variations, although it requires fine-tuning to achieve optimal results [27].

Moreover, Autoencoders, as shown in Figure 1, are architectures based on deep neural networks that compress input data into essential features and then reconstruct the original data from these compressed representations. However, one of the prominent types of these models, known as generative models, is Variational Autoencoders (VAEs). These models learn probabilistic representations of data and generate new samples by altering these representations. VAEs have been employed to create maps and 2D levels for games such as Super Mario Bros and Mega Man [28]. Additionally, methods like Blend-Elites have advanced new content generation by combining features from multiple games. This approach, which integrates VAEs with diversity-quality algorithms (MAP-Elites), successfully generated diverse and playable levels by merging three games. Despite the need for larger and higher-quality datasets, these approaches are considered suitable for PCG due to their ability to generate creative and diverse content [7].

Table 2 separately and concisely summarizes the applications, advantages, and limitations of the studies discussed in this subsection to provide a clearer perspective. Accordingly, CNNs are more suitable for generating high-quality visual content, LSTMs for sequential data, and VAEs for producing diverse content. It is worth mentioning that this subsection only examines some of the common deep learning-based techniques, while more modern models are discussed in subsequent subsections. In general, despite requiring computational resources and large datasets, validation challenges (Table 2), and lacking interpretability [18], these methods have played a significant role in advancing PCG in various domains.

Table 2. An overview of the applications of deep learning techniques in PCG.

Model/Algorithm	Application	Advantages	Disadvantages/challenges	References
CNNs	Game character generation	Quality and production speed	Overfitting and the need for a more complex render model	[23]
CNNs	Predicting balance in asymmetric scenarios in a top-down shooter game, The Rogue-Like game Nuclear Throne	Balance	Need for real player behavior data	[24]
Bidirectional LSTM and fuzzy-GA hierarchical models	Game level generation	Personalized	High computational resources and parameter tuning	[26]
LSTM	Adapted levels generation in Lode Runner game	Improvement of the global consistency of generated levels	Uncertainty about playability	[25]
Generative Flow Model (WaveFlow)	Controllable sound effect generation	Controllable sound variations and style transfer capabilities	Need for fine-tuning and quality enhancement by sound experts	[27]
VAE and random forest classifier	Level generation in SMB, Kid Icarus, and Mega Man	Generation of traversable and combinational levels	Need for ensuring playability and quality of levels and controllability	[28]
VAEs and MAP-Elites algorithms	Game blending on five platformer games	Generation of a diverse range of playable levels	Lack of data	[7]

These include character generation, sound effects, and the creation of combinational, controllable, balanced, personalized, and playable levels. In contrast, traditional ML algorithms, while potentially lacking the same level of accuracy or performance, are generally faster, more interpretable, and require fewer computational resources [18].

3.2. Generative AI and Adversarial Learning

Generative AI, as shown in Figure 1, is an advanced branch of deep learning that focuses on generating new content such as text, images, and game levels. The two main frameworks in this field are Generative Adversarial Networks (GANs) and pre-trained transformers such as Generative Pretrained Transformer (GPT-2) [18]. GANs, through competition between a generative network and a discriminator, are capable of producing complex content such as maps, characters, and game environments. These models are particularly effective in generating pixel-based content or tile arrays, such as maps and game characters, especially in tasks requiring high diversity and complexity. However, they come with limitations such as the need for more training data, instability in results, and challenges in content validation [29]. Additionally, adversarial learning, which refers to competition between different components of a model, has many applications in PCG for games. This approach is particularly useful for generating realistic data and combating fake data in complex games. Furthermore, GAN-based models are also categorized under adversarial learning methods and act as generative AI models [8]. Therefore, as shown in Figure 1, adversarial learning overlaps with generative methods.

Therefore, examples of applications of generative artificial intelligence and adversarial learning include recent research using GAN-based models. For instance, PokerFace-GAN has been used to assess facial similarity and generate game characters [30], and another GAN model has been employed to produce 3D terrains with playability and replayability for video games [31]. Moreover, Schubert et al. [32] introduced a novel one-shot GAN algorithm that generates Super Mario Bros (SMB) levels of any size from a single example, extending it for multi-shot learning by integrating multiple level styles. Additionally, Hald and colleagues [33] employed parametric GANs with conditional vectors to control output details in level generation. Match-3 games have also been used as an environment to evaluate GAN performance in generating complex patterns. Understanding complex and symmetrical visual patterns is a key challenge, and all these patterns require various levels of domain knowledge for identification [34]. Furthermore, Compositional Pattern Producing Networks (CPPNs) are suitable for generating content based on geometry and complex pattern design. Pre-trained GANs and CPPNs have demonstrated good performance in generating levels for games like Super Mario Bros and The

Legend of Zelda and have been effective in covering specific areas that other methods cannot achieve [35]. However, the main limitations of these models include the need for larger datasets and high computational resource consumption.
In contrast, shallow adversarial learning based on traditional ML models can be a suitable choice in scenarios with limited computational resources. Although adversarial learning is often applied using deep learning-based architectures, its concepts are not dependent on any specific model and can be applied with other traditional ML methods such as decision trees [36], as shown in Figure 1. For example, Murer and colleagues [36] applied adversarial methods to generate levels by replacing the fitness function in a Greedy Search System with a random forest classifier to distinguish between human and random levels in games like Frogger, Zelda, and Aliens. It can also be said that adversarial learning, as shown in Figure 1, does not necessarily fall into the generative category, and adversarial methods can be used to identify and counter adversarial samples in this field.
Table 3 provides a summary of the applications, advantages, and disadvantages of the works discussed in this area. Overall, generative models and adversarial learning, especially GANs, play a crucial role in introducing diversity and creativity for generating game levels, combining games, characters, and terrains, despite challenges with result instability, noise [9], the quantity and quality of training data, computational resource and content validation. Moreover, the ability to create creative and diverse content distinguishes these models from other machine learning-based models. Additionally, for scenarios with limited resources, more traditional methods such as CPPNs or other shallow adversarial learning approaches may be a better choice.

3.3. Reinforcement Learning

As mentioned earlier, models such as VAEs and GANs face challenges in game content generation due to the limited availability of training data. In contrast, RL can operate effectively with less data through self-learning and optimization within the Markov Decision Process (MDP) framework, enabling the generation of diverse and personalized content in dynamic and complex gaming environments [37].
Examples of RL-based models include classical RL approaches like Q-Learning, which do not rely on deep neural networks but instead use Q-tables. For instance, Q-Learning has been employed to train a deck-building system where a small network with a single hidden layer is used to generate a card deck capable of defeating another deck tailored to a specific player [38]. Thus, as illustrated in Figure 1, some RL methods do not fall within the realm of deep learning. However, more advanced models like Deep Q-Networks (DQN) and Proximal Policy Optimization (PPO) leverage deep neural networks and are better suited for more complex applications with larger state spaces, such as PCG in video games.
For example, Khalifa et al. [39] used RL to generate challenging levels across three different games, optimizing the gaming experience, albeit at the cost of a time-consuming learning process.

Table 3. Applications of generative AI and adversarial learning techniques in PCG.

Model/Algorithm	Application	Advantages	Disadvantages/challenges	References
One-shot GAN	Level generation in SMB	Combining various styles and levels of different sizes	Requiring high computational resources and risking the generation of fake data	[32]
Parametric GANs	Game level generation	Controllable	Requiring more training data, precise tuning, and addressing balancing challenges	[33]
GAN-based model	3D terrain generation	Diverse, engaging, playable, and replayability content	Requiring more data	[31]
GANs	Generating complex patterns in match-3 games	Understanding complex and symmetrical visual patterns	Requiring domain knowledge	[34]
GANs	Game character generation	Generalization	Requiring more data	[30]
GAN and CPPNs	Level generation in SMB and The Legend of Zelda	Covering spaces better, reproducing global and local patterns in complete levels, and enabling changes during training	Tending to over-repeat blocks	[35]
Adversarial random forest	Level generation in Frogger, Zelda and Aliens games	Human-like quality	Requiring more data	[36]

Similarly, Shaker et al. [40] developed the PCGRL framework to generate endless and diverse levels for SMB based on player experience rewards. However, balancing the reward mechanism posed a significant challenge, occasionally resulting in repetitive content.

Furthermore, Zakaria et al. [41] compared PCGRL and GANs in generating Sokoban levels, demonstrating that PCGRL, with PPO-based agents and Generative Playing Networks (GPNs), provides greater diversity, whereas GANs offer finer control over details. Additionally, the PCGRL Puzzle framework was introduced, utilizing RL and PPO to generate diverse scenarios, offering a personalized experience [42]. Moreover, the ASTER-XT model employed RL to create coherent narratives [43].

In physics-based games like Angry Birds, Gamage et al. [44] designed a framework for generating new levels that ensure novelty and align with design objectives. For turn-based RPGs, algorithms like DQN and Deep Deterministic Policy Gradients (DDPG) with dense rewards were used to enhance level quality and scoring systems [45]. In another study, Wang et al. [46] proposed an RL-based framework for real-time generation of playable music levels. Additionally, Jiang et al. [47] used PPO and neural cellular automata to generate diverse levels in Minecraft. Wang et al. [48] combined pre-trained GANs with SAC-based RL designers to create SMB levels, while Huber et al. [49] introduced a Dynamic Difficulty Adjustment (DDA) approach using deep RL to tailor VR sports game maze structures to players' skill levels. Nonetheless, a common limitation of these approaches is the need for fine-tuning and further evaluations to produce desirable, playable content.

Table 4. Applications of RL techniques in PCG.

Model/Algorithm	Application	Advantages	Disadvantages/challenges	References
PPO based RL	2D levels generation in Binary, Zelda, and Sokoban games	Playable	Time-Consuming Learning Process	[39]
PPO based RL and MarioGAN	Experience-driven and endless levels generation in SMB	Endless, playable, Real-time, personalized, ensuring a certain degree of fun and deviation across level segments	Strong Dependence on Reward Function and Agent Behavior	[40]
Controllable and uncontrollable PCGRL with agents based on PPO and GPN, GANs, VAEs, and VAEGANs	Mixed-initiative cooperative levels generation in Sokoban	Increased playability at the expense of diversity, reduced training time	Lack of Quality, Poor Agent Performance	[41]
PPO based RL	Mixed-initiative level generation in 2D puzzle games	Diverse	Reward Function Adjustment	[42]
DQN based RL	Controllable level generation in Angry Birds, a physics-based video game	The generated novelties are consistent with user-defined objectives	Requirement for Advanced Agents	[44]
DQN based RL, GPT-2	Storyline events generation	Interesting and coherent sentences	Lack of Focus on Emotions	[43]
DQN and DDPG based RL	Level generation in turn-based RPG	Real-time, denser rewards, high-quality, and diverse	Reward Delays, Need for Further Evaluation	[45]
SAC-based RL, GAN and KNN	Real-time level generation from Music in SMB	Playable	Need for Human Experiments	[46]
PPO based RL and neural cellular automata with 3D convolutions	Game maze and dungeon generation in Minecraft game	Controllable	not uniform in shape and size	[47]
SAC based RL and GAN	SMB levels generation	Endless, playable, fun, and diverse	bsence of a Universal Method to Evaluate Fun	[48]
DQN based RL	Experience-driven game maze generation with DDA in virtual reality sports games	Personalized	Need for More Human Evaluations, Failure to Generate Levels with Appropriate Difficulty, Slow Player Adaptation	[49]
DQN based RL	Transfer learning from Mario to Zelda for Zelda dungeon generation	Interaction with human designers	Requirement for Real Interaction Data, Differences in Game State Space Sizes	[10]

Moreover, in the context of transfer learning, applied knowledge obtained from Super Mario to generate dungeons in Zelda. However, challenges such as ensuring the relevance of transferred knowledge and selecting appropriate techniques persist [10].

Table 4 summarizes recent research on RL applications in PCG for games. Overall, unlike models such as CNNs, VAEs, GANs, and LLMs, which face data constraints, RL excels in creating engaging player experiences through reward-based feedback and self-learning. Moreover, despite challenges like reward delays, lengthy learning, and reliance on data or human evaluations for playability, RL effectively delivers interactive, personalized content.

3.4. Diffusion Models

Diffusion models, based on Markov chains, involve adding noise to real data (forward diffusion) and training a model to reverse this process (reverse diffusion) using deep neural networks [50]. These models generate consistent and accurate content aligned with the original data distribution [51].

For example, Dai et al. [51] employed unconditional diffusion models to generate levels for 2D games like SMB and 3D games like Minecraft. Their modifications included replacing one-hot encoding with word embeddings to improve scalability and stability, and using fully convolutional networks with residual connections to mitigate overfitting. This approach highlighted finer details such as house features and reduced visual artifacts like floating blocks compared to GANs. Additionally, the LLMaker model generates game levels while interacting with users through an LLM. This model leverages Stable Diffusion to create graphical assets such as backgrounds, enemies, traps, and treasures based on LLM descriptions [52]. However, these models require extensive datasets, high computational resources, and content evaluation requirements.

Moreover, recent studies have applied denoising diffusion models (DDMs) to point clouds for 3D shape generation [53–55]. While PVD [55] directly trains on point clouds, DPM [53] and LION [54] introduce a latent shape variable into the process. Additionally, the NeuralField-LDM model has been developed for generating open-world 3D scenes. This model represents scenes as dense voxel grids, which are then compressed using a latent autoencoder. Moreover, a hierarchical diffusion model, similar to LION and trained with separate conditional DDMs, is utilized to generate these scenes, supporting conditional scene generation and completion [56]. Challenges for these models include slow processing, complexity in aligning with the original data distribution, and the need for multi-view images.

The applications, advantages, and limitations of these studies are summarized in Table 5. Overall, diffusion models, compared to other methods, improve scalability, stability, and overfitting by iteratively combining noise addition and removal processes. Compared to models like GANs, they enable the generation of more precise and high-quality content, ranging from game levels to 3D shapes and open-world scenes. Nevertheless, these models face challenges such as the need for large datasets and multi-view images to maintain

Table 5. Applications of diffusion models in PCG.

Model/Algorithm	Application	Advantages	Disadvantages/challenges	References
Unconditional Diffusion Models	Level generation from single samples in SMB and Minecraft	Human-like quality	Higher computational resources and time compared to GANs	[51]
Stable Diffusion Models and LLM	Assistance for procedural graphical asset generation (e.g., backdrops, sprites)	Text and image generation, human-AI co-creativity	Need for difficulty evaluation	[52]
Denoising Diffusion Models	3D shape generation (point clouds)	Flexible	Difficulty in modeling point distributions	DPM [53]
		Noise removal for multi-faceted shapes and shape interpolation	Need for texture synthesis	LION [54]
		Diverse and high-quality completion from a single partial shape	Possibility of generating shapes away from the ground truth when dealing with unconventional real-world shapes	PVD [55]
Hierarchical diffusion model similar to LION and trained with separate conditional DDMs	Open-world scene generation	Controllable and high-quality	Slowness, requires multi-view images	[56]

distribution alignment, as well as high computational resource consumption due to their iterative processes.

3.5. NLP and LLMs

NLP is a subfield of ML focused on the analysis and generation of natural language, with applications in chatbots, machine translation, and games [11]. However, some NLP techniques, such as word frequency analysis, regular expressions (regex) [57], classical Named Entity Recognition (NER), and WordNet Synsets, rely on statistical principles [58] and are not considered part of AI [57]. Additionally, until five years ago, autoregressive models in NLP were mostly theoretical. The release of LLMs, particularly the GPT-2 in 2019, demonstrated the capability of transformer-based models in generating coherent text, paving the way for more advanced models like ChatGPT in 2022 [9]. For example, Freiknecht et al. [54] used classical NER, WordNet Synsets, and the GPT-2 language model to create branching and interactive narratives for games. This process utilized a handwritten introduction and predefined goals for story generation. Thus, given the diversity of NLP techniques, as illustrated in Figure 1, NLP methods span a broad range from outside the AI domain to deep learning-based approaches such as LLMs.

LLMs, located at the intersection of NLP and deep learning (as shown in Figure 1), are designed with diverse architectures for various tasks. For example, GPT excels at text generation and interactive responses, while BERT is specialized for text classification and sentiment analysis. Furthermore, these models are not limited to NLP tasks. With capabilities such as multimodal content generation, exemplified by GPT-4V and Llava in processing text and images, and their adaptation to non-textual data, LLMs have revolutionized PCG in games [11].

For instance, in the VOYAGER system, GPT-4 utilized the Mineflayer API in Minecraft to generate complex actions that translated into in-game skills, although its success heavily depended on the robust API [59]. LLMs have also enhanced NPC interactions in games by producing dynamic, context-aware responses that make gameplay more engaging. Park et al. [60], for example, employed LLMs in a Sims-like environment to generate believable behaviors, new objectives, and memory of past events for NPCs. However, such advancements require significant computational resources, additional evaluation, and fine-tuning of relevant functions.

Moreover, LLMs have been employed in platforms like Twitch for game commentary. In games such as League of Legends, LLMs identify key events, and ChatGPT generates commentary styled after fictional characters, with voices synthesized using the FakeYou3 API. However, this approach demands more extensive data collection to encompass a broader range of in-game events and to evaluate the commentary's entertainment value while conveying critical insights [61].

In narrative-driven experiences, LLMs act as game masters. For example, in AI Dungeon game, which is based on a fine-tuned GPT-2, players interact with the story through simple text inputs, and the LLM advances the narrative akin to a human game master. Nevertheless, this capability to create complex storylines requires extensive computational resources and data [62]. Similarly, games have leveraged text-to-image models such as Stable Diffusion to enhance visual storytelling [63].

From a game mechanics perspective, natural language interactions offer a unique feature for LLM-based games. In 1001 Nights game, LLMs generate stories based on player-provided keywords [64]. Additionally, GPT-3 personalizes game narratives by combining user-selected elements, encouraging players to discover specific keywords to create new content based on physical and chemical properties [65]. While this allows players to have a personalized experience, the inherent uncertainty in LLM-generated text may occasionally adversely impact game mechanics [64, 65].

In game content generation, LLMs like GPT-2 have been utilized to design puzzles in games such as Sokoban, although their performance is limited with small datasets [66]. More advanced models like GPT-3 have addressed these challenges by providing procedural assistance or directly refining content based on user requirements. For instance, in designing the Metaweide game, GPT-3 generated game levels based on user-described features, ensuring playability through human supervision [67]. Tools like LLMaker also enable users to interact directly with language models for game level design and refinement [68], enhancing the design process but necessitating additional data and human oversight for ensuring playability.

Table 6 highlights the applications, advantages, and limitations of the discussed studies. In conclusion, compared to other methods, LLMs demonstrate superiority in creating engaging

Table 6. Applications of NLP and LLMs models in PCG.

Model/Algorithm	Application	Advantages	Disadvantages/challenges	References
NER, WordNet Synsets and GPT-2	Branched storyline generation	Story coherence	Overproduction of characters and similar themes	[58]
GPT-4 to interface with the Mineflayer API	Code and actions generation in Minecraft	Robust to model variations	Unachievable tasks, dependent on strong API and fine-tuning	[59]
ChatGPT	Enhancing NPC interactions in a Sandbox environment, reminiscent of The Sims	Believable behavior	High computational resource needs, further evaluation, and tuning	[60]
ChatGPT	Automated commentary on game events on platforms like Twitch in League of Legends.	Meaningful interpretation during live streaming	Requires data collection and manual annotation of gameplay videos	[61]
GPT-2	Continuing the story like a human GM in AI Dungeon, a text-based adventure game	Human-AI collaborative creativity	High computational and data demands	[62]
GPT-2 + Stable Diffusion	Visual storytelling generation in AI Dungeon	Low computational resources while maintaining quality and flexibility	Need for precise model tuning	[63]
GPT-4 and Stable Diffusion	Interactive narrative generation and text-to-image transformation in the 1001 Nights game	Real-time, guaranteed victory in combat	Unpredictability, balancing generated content with game rules	[64]
GPT-3	Text-based game scenarios generation	Real-time, high logic, repeatable playability, and more content	Potential negative impacts on game mechanics due to text uncertainty	[65]
GPT-2	level generation to create new puzzles in Sokoban	A diverse set of new and playable levels	Insufficient data and training	[66]
LLMaker is powered by GPT-3.5-Turbo1106, Stable diffusion models	Chat-based level editor to adjust level design	Real-time, Co-creativity based on natural language,	Risk of unplayable or imbalanced levels	[68]
GPT-3	Game level generation in Metaweidel	Increased playability, efficient use of limited data without overfitting	Limited accessible data	[67]

textual interactions and narratives due to their ability to analyze and generate natural language.
Their pretrained nature enables them to independently create diverse and complex game content, even in real-time. Despite challenges like reproducibility, data biases, and high computational demands, LLMs are promising tools for innovation in the gaming industry as creative collaborators and design assistants.

4. Discussion and Open Problems in PCG

Figure 2 provides an overview of the applications of various ML techniques in PCG in games, based on the studies reviewed in this research. As shown, classical ML approaches have been applied in areas such as image generation, music composition, and gameplay mechanics. However, as discussed in Section 3, these approaches do not directly contribute to content generation. Their utility lies in their ability to predict and classify specific tasks with well-defined features and limited, structured data, making them suitable for simpler and faster tasks.

In contrast, deep learning, despite its computational complexity and higher demand for data and resources, plays a direct role in generating levels, characters, and music, as illustrated in Figure 2. This is due to its superior ability to extract complex features from unstructured and extensive datasets. While all references in other parts of the figure rely on deep neural networks, the contributions of deep learning in this figure, as outlined in Section 3.1, are primarily attributed to the use of CNNs for generating high-quality visual content, LSTMs for sequential data, and VAEs for producing diverse content, particularly in level generation.

RL, as shown in Figure 2, is used for story creation and, more notably, for game level generation. Because, RL allows the creation of interactive, personalized content through reward feedback, providing better management of dynamic game scenarios than other methods. However, challenges like reward delays, lengthy learning processes, and the need for human interaction data to ensure playability still pose significant obstacles.

Additionally, NLP models and LLMs have been utilized for generating game levels, narratives, and stories, as well as for gameplay commentary and

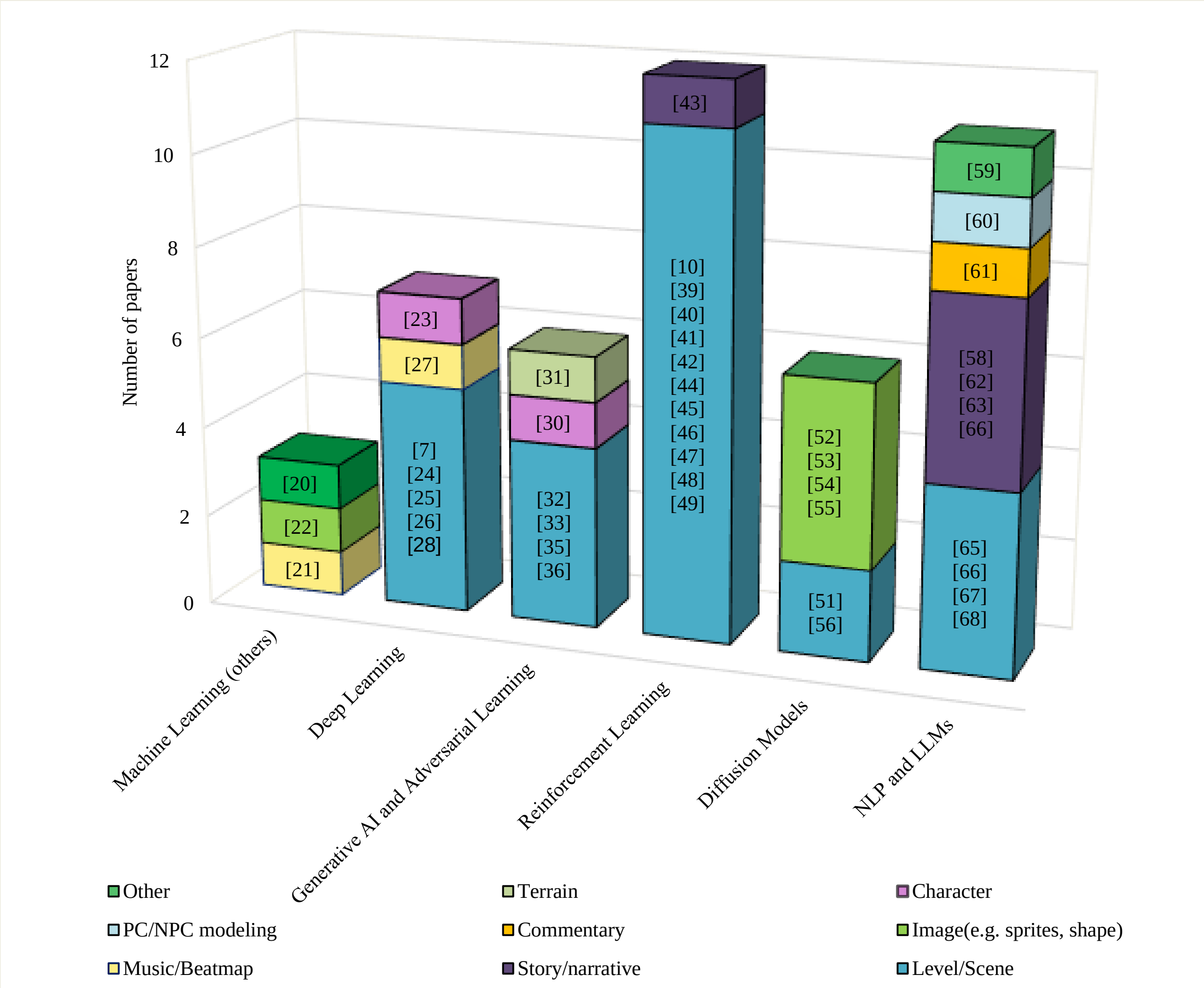

Figure 2. The frequency of reviewed studies by machine learning methods and application types in game content generation.

player dialogue modeling. Due to their natural language understanding and generation capabilities, particularly with the use of pre-trained LLMs, NLP models excel in generating natural dialogues, compelling storylines, and real-time textual interactions compared to other methods. However, challenges such as the need for large training datasets, higher computational resources, and uncertainty about the quality of generated content due to the inherent unpredictability of LLMs persist.

Moreover, diffusion models, due to their ability to produce detailed and high-quality content, have been more extensively applied in image generation and, subsequently, in level design, as shown in Figure 2. In contrast, generative adversarial models (primarily GANs, as discussed in Section 3.4) are preferred for game level generation because of their ability to create diverse and creative content, which is particularly valuable for game levels. This feature has also led to their application in designing game characters. However, the primary limitation of these models is their dependency on high-quality input data and their high computational costs.

In conclusion, modern ML approaches are increasingly being used compared to classical ML and deep learning techniques. Each approach offers specific capabilities, enabling more applications in a particular field. However, ML encompasses a wide range of approaches, and even for a single task, multiple methods or their combinations may be available [9].

Moreover, PCG in games faces several challenges that impact its effectiveness and efficiency. Notable issues include the scarcity of training data, learning from small datasets, style transfer, and the need for precise metrics to evaluate the quality of generated content [6,8,12-15]. Additionally, parameter tuning remains a critical issue in PCG. However, as discussed earlier, ML approaches can partially address these challenges. Figure 3 classifies the most significant challenges identified in this article.

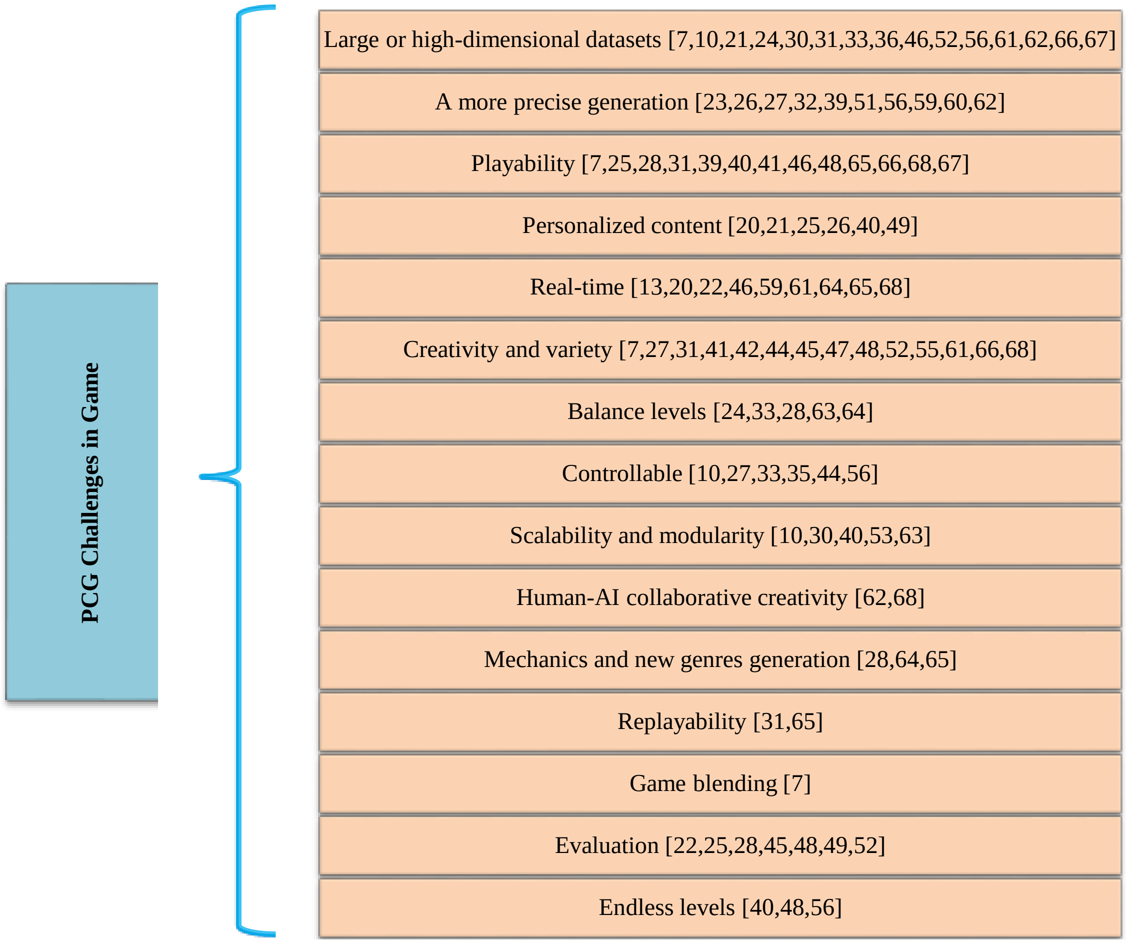

Figure 3. Taxonomy of the PCG challenges in games.

Deep learning and RL techniques have significantly advanced PCG by improving the generation of images, sounds, and game levels. Nevertheless, they continue to face challenges in maintaining content quality and coherence, as well as in managing the substantial computational resources required [8,25,26]. Adversarial learning techniques, such as GANs, face challenges related to training data quality, playability, and generating detailed and diverse content [7,31,34,36].
Diffusion models, which are increasingly used in PCG, encounter challenges like scalability, stability, and generating fine details while minimizing visual artifacts [51-53]. LLMs face difficulties with generalizability to less popular games [59], handling small training datasets [66], and directly creating and modifying game content [11]. Moreover, human oversight is often needed to ensure playability [67], and success heavily relies on access to advanced technologies and resources [59].
Despite these challenges, ongoing research in PCG is exploring various strategies to overcome existing problems. However, more work is needed to address the unique challenges of creating game content and ensure that algorithmic decision-making remains unbiased and effective.

5. Conclusion

The application of advanced ML techniques in PCG presents exciting opportunities for creating diverse and engaging game content. Techniques such as adversarial learning, RL, NLP, and diffusion models have significantly advanced various aspects of PCG, including level design and narrative creation. Moreover, efforts to personalize games, maintain balance, and ensure playability are of utmost importance.
Despite these advancements, challenges related to data requirements, computational costs, model generalization, and content quality persist. Additionally, future research should focus on addressing less-explored challenges such as style transfer, generating new game genres, and ensuring unbiased algorithmic decision-making. By overcoming these challenges, the field of PCG can deliver richer and more dynamic gaming experiences.

References

[1] L. Rodrigues, B. Robson and J. Brancher, "Procedural versus human level generation: Two sides of the same coin?," *International Journal of Human-Computer Studies*, vol. 141, p. 102465, 2020.

[2] A. Latif, M.F. Zuhairi, F.Q., P. Khan Randhawa and A. Patel, "A critical evaluation of procedural content generation approaches for Digital Twins," *Journal of Sensors*, 2022(1), p.5629645, 2022.

[3] F. Quarneti, "Steam Sets New Concurrent Player Record with Over 37 Million Users," [Online]. Available:https://www.benzinga.com/general/gaming/24/08/40560450/steam-sets-new-concurrent-player-record-with-over-37-million-users. [Accessed: August. 26, 2024].

[4] G. Smith, "Understanding procedural content generation: a design-centric analysis of the role of PCG in games," *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 917-926, 2014.

[5] M. Hafis, H. Tolle, and A. Afif Supianto, "A literature review of empirical evidence on procedural content generation in game-related implementation," *Journal of Information Technology and Computer Science.* vol. 4, pp. 308-328, 2019.

[6] A. Summerville, S. Snodgrass, M. Guzdial, C. Holmg˚ard, A. K. Hoover, A. Isaksen, A. Nealen, and J. Togelius, "Procedural content generation via machine learning (pcgml)," *IEEE Transactions on Games,*.vol. 10, pp. 257–270, 2018.

[7] A. Sarkar and S. Cooper, "Generating and blending game levels via quality-diversity in the latent space of a variational autoencoder," *in The 16th International Conference on the Foundations of Digital Games (FDG)*, pp. 1–11, 2021.

[8] J. Liu, S. Snodgrass, A. Khalifa, S. Risi, G. N. Yannakakis, and J. Togelius, "Deep learning for procedural content generation," *Neural Computing and Applications. 33(1)*, pp. 19-37, 2021.

[9] Mao, X., Yu, W., Yamada, K.D. and M.R. Zielewski, "Procedural Content Generation via Generative Artificial Intelligence," *arXiv preprint arXiv:2407.09013*, 2024.

[10] Z. Zhou and M. Guzdial, "Toward co-creative dungeon generation via transfer learning," *in the 16th International Conference on the Foundations of Digital Games (FDG)*, pp. 1–9, 2021.

[11] R. Gallotta, G. Todd, M. Zammit, S. Earle, A. Liapis, J. Togelius, and GN. Yannakakis, "Large language models and games: A survey and roadmap," *arXiv preprint arXiv:2402.18659*. 2024.

[12] J. Freiknecht and W. Effelsberg, "A survey on the procedural generation of virtual worlds," *Multimodal Technologies and Interaction*, 1(4), p.27, 2017.

[13] R.L. Cabrera, M.N. Collazo, C.C. Porras and A.J.F. Leiva, "Procedural content generation for real-time strategy games," *IJIMAI*, vol. 3, pp. 40-48, 2015.

[14] N.A. Barriga, "A short introduction to procedural content generation algorithms for videogames," *International Journal on Artificial Intelligence Tools*, 28(02), p.1930001, 2019.

[15] S. Risi, and J. Togelius, "Increasing generality in machine learning through procedural content generation," *Nature Machine Intelligence*, 2(8), pp.428-436, 2020.

[16] P. Sweetser, "Large language models and video games: A preliminary scoping review," *InProceedings of the 6th ACM Conference on Conversational User Interfaces*, pp. 1-8, 2024.

[17] M. Hendrikx, S. Meijer, J. Van Der Velden, A. Iosup, "Procedural content generation for games: A survey," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 9(1):1-22. 2013.

[18] L.P. Zhuhadar and M.D. Lytras, "The application of AutoML techniques in diabetes diagnosis: current approaches, performance, and future directions," *Sustainability*, *15*(18), p.13484, 2023.

[19] Y. Wu, A. Yi, C. Ma, and L. Chen, "Artificial intelligence for video game visualization, advancements, benefits and challenges," *Mathematical Biosciences and Engineering*, 20(8), pp.15345-15373, 2023.

[20] J. Liang, Y. Tang, R. Hare, B. Wu, and F.Y. Wang, "A learning-embedded attributed Petri net to optimize student learning in a serious game," *IEEE Transactions on Computational Social Systems*, 10(3), pp.869-877. 2021

[21] E.A.L. Estolas, A.F.V. Malimban, J.T. Nicasio, Rivera, J.S., M.F.D. San Pablo, and T.L. Takahashi, "Automatic Beatmap Generating Rhythm Game Using Music Information Retrieval with Machine Learning for Genre Detection," *In IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*. pp 1-6, 2020.

[22] J. Rafner, S. Langsford, A. Hjorth, M. Gajdacz, L. Philipsen, S. Risi, J. Simon, J. Sherson, "Utopian or Dystopian?: using a ML-assisted image generation game to empower the general public to envision the future," *In Creativity and Cognition*, pp 1-5, 2021.

[23] T. Shi, Z. Zou, Z. Shi, and Y. Yuan, "Neural rendering for game character auto-creation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(3), pp.1489-1502, 2020.

[24] C. Stephens and DC. Exton, "Assessing Multiplayer Level Design Using Deep Learning Techniques," *In Proceedings of the 15th International*

Conference on the Foundations of Digital Games, pp 1-3, 2020.

[25] K. Sorochan, J. Chen, Y. Yu, M. Guzdial, "Generating Lode Runner Levels by Learning Player Paths with LSTMs," *In Proceedings of the 16th International Conference on the Foundations of Digital Games*, pp 1-7, 2021.

[26] M. Ince, "BiLSTM and dynamic fuzzy AHP-GA method for procedural game level generation," *Neural Computing and Applications*, 33(15):9761-73, 2021.

[27] S. Andreu and M.V. Aylagas, "Neural synthesis of sound effects using flow-based deep generative models," *In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Vol. 18, No. 1, pp. 2-9, 2022.

[28] A. Sarkar and S. Cooper, "Sequential segment-based level generation and blending using variational autoencoders," *In Proceedings of the 15th International Conference on the Foundations of Digital Games*, pp. 1-9, 2020.

[29] L. Lazaridis and G.F. Fragulis, "Creating a Newer and Improved Procedural Content Generation (PCG) Algorithm with Minimal Human Intervention for Computer Gaming Development," *Computers*, 13(11), p.304, 2024.

[30] T. Shi, Z. Zou, X. Song, Z. Song, C. Gu, C. Fan, and Y. Yuan, "Neutral face game character auto-creation via pokerface-gan," *In Proceedings of the 28th ACM International Conference on Multimedia*, pp. 3201-3209, 2020.

[31] G. Ivanov, MH. Petersen, K. Kovalský, K. Engberg, G. Palamas, "An Explorative Design Process for Game Map Generation Based on Satellite Images and Playability Factors," *In Proceedings of the 15th International Conference on the Foundations of Digital Games*, pp. 1-4, 2020.

[32] F. Schubert, M. Awiszus, and B. Rosenhahn, "Toad-gan: a flexible framework for few-shot level generation in token-based games," *IEEE Transactions on Games*, 14(2):284-93, 2021.

[33] A. Hald, JS. Hansen, J. Kristensen, and P. Burelli "Procedural content generation of puzzle games using conditional generative adversarial networks," *In Proceedings of the 15th International Conference on the Foundations of Digital Games*, pp. 1-9, 2020.

[34] V. Volz, N. Justesen, S. Snodgrass, S. Asadi, S. Purmonen, C. Holmgård, J. Togelius, and S. Risi, "Capturing local and global patterns in procedural content generation via machine learning," *In IEEE Conference on Games (CoG)*, pp 399-406, 2020.

[35] J. Schrum, B. Capps, K. Steckel, V. Volz, and S. Risi, "Hybrid Encoding for Generating Large Scale Game Level Patterns with Local Variations," *IEEE Transactions on Games*, 2022.

[36] T. Maurer and M. Guzdial, "Adversarial Random Forest Classifier for Automated Game Design," *In Proceedings of the 16th International Conference on the Foundations of Digital Games*. pp 1-6, 2021.

[37] S. Nam and K. Ikeda, "Generation of diverse stages in turn-based role-playing game using reinforcement learning", *In 2019 IEEE Conference on Games (CoG)*, pp. 1-8, 2019.

[38] Z. Chen, C. Amato, T.H.D. Nguyen, S. Cooper, Y. Sun, and M.S. El-Nasr, "Q-deckrec: A fast deck recommendation system for collectible card games," *In 2018 IEEE conference on Computational Intelligence and Games (CIG)*, pp. 1-8, 2018.

[39] A. Khalifa, P. Bontrager, S. Earle, J. Togelius, "Pcgrl: Procedural content generation via reinforcement learning," *In Proceedings of the AAAI Conference on Artificial Intelligence 2020 and Interactive Digital Entertainment*, 16(1):95-101.

[40] T. Shu, J. Liu, G.N. Yannakakis, "Experience-driven PCG via reinforcement learning: A Super Mario Bros study," I*n 2021 IEEE Conference on Games (CoG)*, pp 1-9, 2021.

[41] Y. Zakaria, M. Fayek, and M Hadhoud, "Procedural level generation for sokoban via deep learning: An experimental study," *IEEE Transactions on Games*, 2022.

[42] P.V.M. Dutra, S.M. Villela, and R.F. Neto, "A mixed-initiative design framework for procedural content generation using reinforcement learning," *Entertainment Computing*, 52, p.100759, 2025.

[43] L.J. Martin, "Neurosymbolic Automated Story Generation," Doctoral dissertation, Georgia Institute of Technology, Atlanta, GA, USA., 2021.

[44] C. Gamage, V. Pinto, C. Xue, M. Stephenson, P. Zhang, and J. Renz, "Novelty generation framework for AI agents in angry birds style physics games," *In 2021 IEEE Conference on Games (CoG)*, pp 1-8, 2021.

[45] S.G. Nam, C.H. Hsueh, and K. Ikeda, "Generation of Game Stages with Quality and Diversity by Reinforcement Learning in Turn-Based RPG," *IEEE Transactions on Games*, 14(3):488-501, 2021.

[46] Z. Wang, J. Liu, "Online Game Level Generation from Music," *In 2022 IEEE Conference on Games (CoG)*, pp. 119-126, 2022.

[47] Z. Jiang, S. Earle, M. Green, J. Togelius, "Learning Controllable 3D Level Generators," *In Proceedings of the 17th International Conference on the Foundations of Digital Games*, pp. 1-9, 2022.

[48] Z. Wang, J. Liu, and G.N. Yannakakis, "The Fun Facets of Mario: Multifaceted Experience-Driven PCG via Reinforcement Learning," *In Proceedings of the 17th International Conference on the Foundations of Digital Games*, pp 1-8, 2022.

[49] T. Huber, S. Mertes, S. Rangelova, S. Flutura, and E. André, "Dynamic difficulty adjustment in virtual reality exergames through experience-driven procedural content generation," *In2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1-8, 2021.

[50] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, 33, pp.6840-6851, 2020.

[51] M. Zhang, Z. Cai, L. Pan, F. Hong, X. Guo, L. Yang, and Z. Liu, "Motiondiffuse: Text-driven human motion generation with diffusion model," *arXiv preprint arXiv:2208.15001*, 2022.

[52] R. Gallotta, A. Liapis, and G. Yannakakis, "LLMaker: A Game Level Design Interface Using (Only) Natural Language," 2024.

[53] S. Luo, and W. Hu, "Diffusion probabilistic models for 3d point cloud generation," *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2837-2845, 2021.

[54] A. Vahdat, F. Williams, Z. Gojcic, O. Litany, S. Fidler, and K. Kreis, "Lion: Latent point diffusion models for 3d shape generation," *Advances in Neural Information Processing Systems*, 35, pp.10021-10039, 2022.

[55] L. Zhou, Y. Du, and J. Wu, "3d shape generation and completion through point-voxel diffusion," In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5826-5835, 2021.

[56] S.W. Kim, B. Brown, K. Yin, K. Kreis, K. Schwarz, D. Li, R. Rombach, A. Torralba, and S. Fidler, "Neuralfield-ldm: Scene generation with hierarchical latent diffusion models," *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8496-8506, 2023.

[57] D. Jurafsky and J. H. Martin, "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models (3rd Edition)," 2024. [E-book] Available: https://web.stanford.edu/~jurafsky/slp3/.

[58] J. Freiknecht and W. Effelsberg, "Procedural generation of interactive stories using language models," *In Proceedings of the 15th International Conference on the Foundations of Digital Games*, pp. 1-8, 2020.

[59] G. Wang, Y. Xie, Y. Jiang, , A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar, "Voyager: An open-ended embodied agent with large language models," *arXiv preprint arXiv:2305.16291*, 2023.

[60] J.S. Park, J. O'Brien, C.J. Cai, M.R. Morris, P. Liang, and M.S. Bernstein, "Generative agents: Interactive simulacra of human behavior," In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1-22, 2023.

[61] N. Ranella, and M. Eger, "Towards Automated Video Game Commentary Using Generative AI," *In EXAG@ AIIDE*, 2023.

[62] M. Hua and R. Raley, "Playing With Unicorns: AI Dungeon and Citizen NLP," *DHQ: Digital Humanities Quarterly*, *14*(4), 2020.

[63] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684-10695, 2022.

[64] Y. Sun, Z. Li, K. Fang, C.H. Lee, and A. Asadipour, "Language as reality: a co-creative storytelling game experience in 1001 nights using generative AI," In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Vol. 19, No. 1, pp. 425-434, 2023.

[65] L. Huang, and X. Sun, "Create ice cream: Real-time creative element synthesis framework based on gpt3. 0," *In 2023 IEEE Conference on Games (CoG)*, pp. 1-4, 2023.

[66] G. Todd, S. Earle, M.U. Nasir, M.C. Green, and J. Togelius, "Level generation through large language models," In *Proceedings of the 18th International Conference on the Foundations of Digital Games*, pp. 1-8, 2023.

[67] M.U. Nasir and J. Togelius, "Practical PCG through large language models," In *2023 IEEE Conference on Games (CoG)*, pp. 1-4, 2023.

[68] R. Gallotta, A. Liapis, and G. Yannakakis, "Consistent Game Content Creation via Function Calling for Large Language Models," antoniosliapis.com, 2024.

به‌کارگیری یادگیری ماشین برای تولید محتوای رویه‌ای در بازی‌ها: یک مرور جامع

شقایق صفاری، مرتضی دُرّی‌گیو* و فرزین یغمایی

دانشکده مهندسی برق و کامپیوتر، دانشگاه سمنان، سمنان، ایران.

ارسال ۲۰۲۴/۰۹/۱۵؛ بازنگری ۲۰۲۴/۱۲/۰۸؛ پذیرش ۲۰۲۵/۰۱/۰۸

چکیده:

تولید محتوای رویه‌ای (PCG) از طریق تولید خودکار و الگوریتمی محتوا، یکی از حوزه‌های فعال تحقیقاتی در صنعت بازی‌سازی است. در سال‌های اخیر، رویکردهای یادگیری ماشین (ML) نقش اساسی در پیشرفت این حوزه ایفا کرده‌اند. در حالی که مطالعات اخیر، عمدتاً بر بررسی یک یا چند رویکرد خاص در PCG متمرکز بوده‌اند، این مقاله با ارائه دیدگاهی جامع‌تر، طیف گسترده‌تری از رویکردها، کاربردها، مزایا و معایب آن‌ها را بررسی می‌کند. علاوه بر این، چالش‌های کنونی و روندهای بالقوه آینده در این حوزه مورد بحث قرار می‌گیرند. اگرچه این مقاله به دلیل رشد سریع و گسترده این حوزه، قصد ارائه یک مرور کامل از تمام تحقیقات موجود را ندارد، اما تحلیل آن مبتنی بر مقالات منتخب منتشر شده بین سال‌های ۲۰۲۰ تا ۲۰۲۴ است.

کلمات کلیدی: تولید محتوای رویه‌ای، یادگیری ماشینی، هوش مصنوعی.