Using Unconditional Diffusion Models in Level Generation for Super Mario Bros.

Abstract

This study introduces a novel methodology for generating levels in the iconic video game Super Mario Bros. using a diffusion model based on a UNet architecture. The model is trained on existing levels, represented as a categorical distribution, to accurately capture the game's fundamental mechanics and design principles. The proposed approach demonstrates notable success in producing high-quality and diverse levels, with a significant proportion being playable by an artificial agent. This research emphasizes the potential of diffusion models as an efficient tool for procedural content generation and highlights their potential impact on the development of new video games and the enhancement of existing games through generated content.

1 Introduction

The ever-evolving landscape of the video game industry has led to the development of increasingly complex and engaging games that captivate players worldwide. A crucial aspect of game development is generating innovative and challenging levels that provide rich and immersive gaming experiences. Procedural content generation (PCG) has emerged as a potent technique for crafting such levels, automating the design process while preserving diversity and intricacy [1]. PCG refers to the algorithmic creation of game content, such as levels, characters, or storylines, using rules or procedures that ensure variety and adaptability [1]. Among numerous approaches to PCG, employing deep neural networks (DNNs) has shown immense potential in generating high-quality and captivating content [2]. This study investigates the application of diffusion models, a largely uncharted deep learning method within the realm of PCG, for level generation in the iconic video game Super Mario Bros. (SMB).

SMB, launched in 1985, holds a prominent position as a culturally and historically significant game. Its straightforward yet enthralling gameplay, centered around platforming mechanics, tile-based level design, and varied game elements, persistently attracts researchers in PCG and artificial intelligence [3]. Numerous PCG techniques have been utilized to generate levels for SMB, including genetic algorithms [4], Markov chains [5], and DNNs [6, 7, 8, 9]. Nevertheless, the quest for more efficient and productive methods to produce high-quality, engaging levels persists.

Recently, diffusion models have garnered substantial interest in the deep learning community due to their capacity to generate realistic and high-quality samples across various domains, such as images, text, and audio [10]. These models function by modeling the data generation process as a continuous diffusion process, capturing the underlying structures and patterns present in the input data [11]. This approach offers the potential for more stable and accurate content generation compared to other DNNs, for instance, generative adversarial networks (GANs) [12]. Although the application of diffusion models within the context of PCG in video games remains unexplored, their content generation capability in other domains suggests they may offer a promising alternative to existing techniques.

This study applies an unconditional diffusion model to level generation for SMB, utilizing a categorical distribution representation to model the data. We train the model on a dataset of existing levels, enabling it to accurately capture the essential game mechanics and design principles. By evaluating the generated levels in terms of quality, diversity, and playability, we aim to demonstrate the effectiveness of diffusion models as a novel approach to PCG in video games.

This research holds substantial implications for the academic and game development communities. By introducing diffusion models as a potential tool in PCG, our work expands the current understanding of deep learning techniques applied to game design. Furthermore, the successful application of diffusion models in level generation may influence the development of new video games and the enhancement of existing games by providing a new avenue for content generation. Ultimately, we hope the insights gained from this study will inspire further research on diffusion models and their potential applications in the rapidly evolving domain of video games and artificial intelligence.

2 Related Work

This section reviews relevant literature on diffusion models and PCG employing deep learning, focusing on SMB level generation.

2.1 Diffusion Models

Sohl-Dickstein et al. [11] laid the foundation for diffusion models by proposing a deep unsupervised learning approach based on nonequilibrium thermodynamics. Their approach involved training deep networks using diffusion processes, allowing for the discovery of hierarchical structures in the input data. This work pioneered the use of diffusion processes in deep learning.

Ho et al. [10] introduced Denoising Diffusion Probabilistic Models (DDPM), which utilized a continuous diffusion process to model the data generation process. The approach captured the underlying structures and patterns in the input data, resulting in stable and accurate content generation. Their model was conditioned on input data and a noise schedule and used a series of denoising steps to reverse the diffusion process.

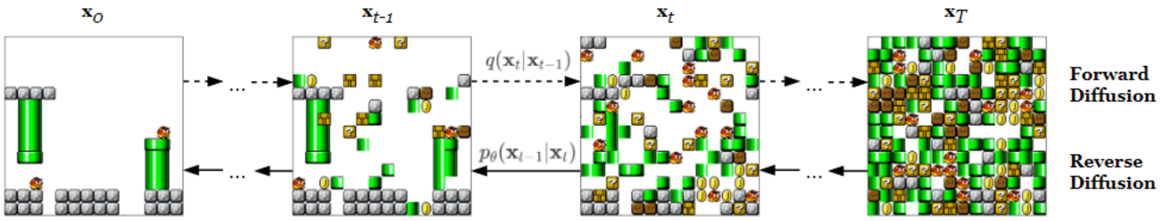

Figure 1: The directed graphical model: forward and reverse diffusion

Dhariwal and Nichol [12] demonstrated that diffusion models could outperform Generative Adversarial Networks (GANs) in image synthesis, further solidifying the potential of diffusion models in content generation. The authors proposed a new sampling algorithm, which reduced the required number of denoising steps and made the generation process more efficient.

2.2 PCG employing Deep Learning Techniques

Summerville and Mateas [6] utilized Long Short-Term Memory (LSTM) networks to generate levels for SMB, treating them as strings and demonstrating the potential of recurrent neural networks in PCG. The LSTM-based approach could capture long-range dependencies in level design but was limited in its ability to represent spatial relationships between game elements.

Volz et al. [7] employed Deep Convolutional Generative Adversarial Networks (DCGANs) to evolve Mario levels in the latent space. This approach allowed for the generation of diverse and engaging content by leveraging the adversarial training process of GANs. Nonetheless, DCGANs could suffer from mode collapse and training instability, leading to a limited variety of generated levels.

Sarkar and Cooper [8] explored the use of Variational Autoencoders (VAEs) for sequential segment-based level generation and blending. The VAE-based approach provided a compact and continuous latent representation of level segments, enabling smooth level generation and blending. One limitation of VAEs was the potential for blurry or less detailed content due to the minimization of reconstruction loss.

More recently, Sudhakaran et al. [9] proposed MarioGPT, a fine-tuned GPT-2 model specifically designed for generating tile-based game levels, focusing on SMB as a use case. The study demonstrated the potential of combining Large Language Models (LLMs) with diversity-driven algorithms like novelty search for open-ended content generation. Despite limitations in generalizability, MarioGPT offered a promising approach to controllable and diverse PCG systems.

While existing literature illustrates some successful implementations of different deep learning techniques in level generation, the use of diffusion models in the realm of PCG remains largely unexplored.

3 Proposed Approach

3.1 Data Collection and Representation

Our training data consists of levels from the Video Game Level Corpus (VGLC) [13], with a focus on two variations of the game: SMB 1 and 2 (Japan). While raw image data of the levels are present, VGLC also provides the levels in the form of long text files, representing different level components with unique characters and necessitating preliminary processing. Discrepancies exist between the two versions, such as inconsistent level heights, varying numbers of unique sprites, and divergent encoding formats. We apply a series of uniformization procedures to resolve the inconsistencies. Subsequently, We partition the standardized text files into uniformly-shaped segments by employing a rolling window. This approach results in level segments measuring 14x14 units and containing 11 distinct sprites. We further encode these segments and transform them into arrays with dimensions of 11x14x14 using one-hot encoding. We represent the levels as a categorical distribution where the 11 unique tokens (sprites) are considered and the probability of each token occurring in a level is modeled.

3.2 Model Architecture

We present an adaptation of the unconditional diffusion model based on a UNet [14] architecture with self-attention mechanisms and temporal embedding.

Self-Attention The self-attention mechanism is a vital component of our model, improving its capacity to capture spatial and temporal dependencies within the input data. Our architecture integrates Performer [15] self-attention layers leveraging the Fast Attention Via positive Orthogonal Random features (FAVOR+) algorithm to reduce the quadratic complexity of self-attention to linear complexity. Although Performer architectures are more commonly applied to natural language processing tasks, recent advancements in transformer-based models like Vision Transformers (ViT) [16] and DETR (DEtection TRansformer) [17] show the potential for using transformers, including Performer, in image-related tasks.

The placement of self-attention layers after each downsampling and upsampling layer enables the model to process long-range dependencies and maintain spatial coherence throughout the network. By leveraging Performer self-attention, the model can effectively learn and utilize both local and global contextual information, leading to improved performance in spatio-temporal prediction tasks while maintaining lower memory usage and computational cost.

Double Convolution The double convolution module consists of two consecutive convolutional layers, each followed by a group normalization [18] layer and a Gaussian Error Linear Unit (GELU) [19] activation function. Group normalization maintains stable distributions of the activations, improving the overall training stability. The GELU activation function introduces nonlinearity to the model, enhancing its capacity to learn complex patterns in the input data.

This combination of layers extracts and processes both high-level and low-level features from the input data, effectively preserving spatial coherence and contributing to improved performance in spatio-temporal prediction tasks. Furthermore, a residual connection can be optionally introduced between the input and output of the module to enhance the model's learning capability and facilitate gradient flow during backpropagation.

Down and Up Blocks The Down and Up blocks are responsible for encoding and decoding the input features, respectively. The Down block consists of a max-pooling layer followed by double convolution layers. Additionally, an embedding layer is used to incorporate the time information $\boldsymbol{t}$ into the Down block. The Up block uses transposed convolution layers that ensure the appropriate upsampling of the input tensor. The Up block concatenates the skip connection from the corresponding Down block and applies double convolution layers. Similar to the Down block, an embedding layer is employed to incorporate the time information $\boldsymbol{t}$ into the Up block.

UNet The UNet is a key component of the proposed model, employing a symmetric encoder-decoder architecture to facilitate the learning of spatio-temporal features in the input data. The model begins with an initial double convolution module to process the input, followed by a series of Down blocks, each incorporating Performer self-attention layers for improved spatial and temporal context representation. The bottleneck is composed of three double convolution modules, designed to capture high-level abstractions. The decoding path consists of a series of Up blocks, again with Performer self-attention layers, enabling the model to recover spatial details from the compressed feature representation. The final convolutional layer maps the output to the desired number of channels. Throughout the architecture, positional encoding is employed to incorporate time information, allowing the model to effectively capture temporal dependencies for improved prediction performance in spatio-temporal tasks.

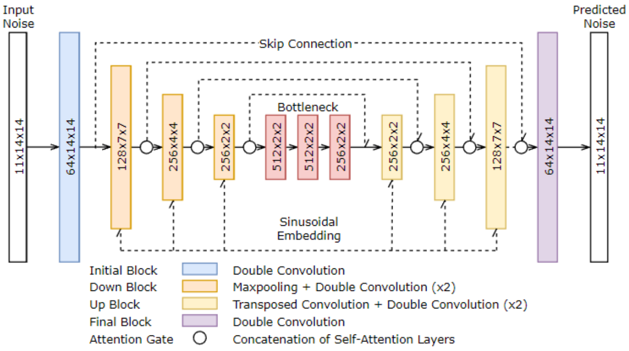

Figure 2: UNet architecture

3.3 Training Process

The training process for our model is inspired by the training and sampling algorithms of DDPM. The core components of the training process include defining a noise schedule, applying forward diffusion to corrupt data samples, training a neural network to perform reverse diffusion by learning a denoising function, and generating new samples by iteratively applying the learned function [10]. We adapt these components to suit the task of level generation for SMB. While maintaining the overall training flow, our model adopts novel approaches and methods tailored to the unique requirements of our task.

4 Experiments

4.1 Methodology

Our proposed model incorporates several novel components that distinguish it from DDPM.

- **Categorical data representation:** We represent the levels using one-hot encoding, enabling the model to learn and generate levels as categorical data. This representation is more suitable for discrete-level generation tasks and allows us to utilize a multi-class cross-entropy loss function.
- **Reconstruction loss:** To address the issue of preserving low-level details, we introduce a reconstruction loss term during the training process, encouraging the model to generate levels with greater fidelity to the original structure. The reconstruction loss is implemented as a negative log-likelihood, which, due to the categorical data representation, essentially becomes a multi-class cross-entropy loss. This loss function, with its equation as below, assists the model in learning effective denoising functions by calculating the negative log probabilities of the original levels given the reconstructed levels.

  $$L_{\text{rec}} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{H}\sum_{j=1}^{W}\log P_\theta(O_{nij}|x_{nij})$$

  N represents the batch size, H and W the height and width of the levels, x_{nij} and O_{nij} the generated and original blocks, respectively, at position

(i, j) in the n-th sample, and $P_\theta(O_{nij}|x_{nij})$ the probability of the original block given the generated block under the parameterized model.

- **Beta scheduling schemes:** We experiment with different beta scheduling schemes, specifically linear, quadratic, and sigmoid scheduling, to investigate their impact on the quality and diversity of the generated levels.
- **Per-sprite temperature scaling:** To balance the representation of different sprite types and better control the sampling of individual sprites, we employ per-sprite temperature scaling and temperature adjustment. Temperature adjustment is achieved by taking the n-th root of the sprite counts and normalizing each by dividing by the count of the least frequent sprite. We experiment with three distinct values of n: 2, 4, and 8. Lower values of n promote *coarse* temperature adjustments, resulting in high-variation samples, while higher values of n encourage *fine* temperature adjustments, leading to low-variation samples.

4.2 Results and Evaluation

4.2.1 Qualitative Evaluation

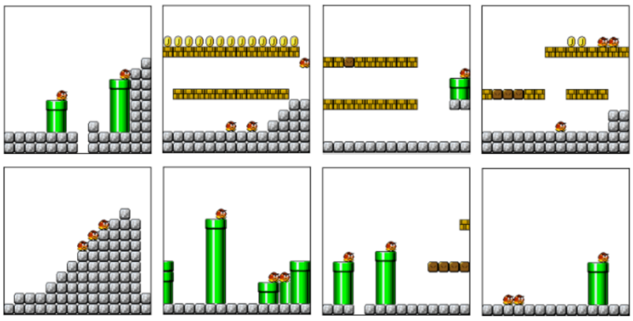

Figure 3: Random levels generated

Figure 3 shows a random sampling of levels utilizing quadratic scheduling and moderate temperature scaling, which led to the fastest convergence and the lowest overall loss after training for 500 epochs. The sampled levels appear to accurately capture SMB's design principles and are mostly hard to distinguish from existing levels using human visual perception.

4.2.2 Quantitative Evaluation

We evaluate the diversity and playability of two versions of our approach, quadratic beta scheduling with moderate and coarse temperature scaling. We set MarioGPT as our baseline. The metrics are average edit distance between pairs of levels, coverage, which is the proportion of levels having another level within a specified distance threshold, and playability, which is the proportion solvable by an A* artificial agent. Table 1 reveals that coarse temperature scaling achieves the highest diversity, having the largest average edit distance and lowest coverage value, although with the cost of reduced playability. The moderate temperature scaling approach strikes a more balanced compromise between diversity and playability, with only slightly lower playability than the baseline (70% vs. 74%).

Table 1: Qualitative Evaluation

Metrics	Moderate temp. scale	Coarse temp. scale	MarioGPT
Edit distance	41.89	61.89	50.78
Coverage	0.33	0.02	0.2
Playability	0.70	0.62	0.74

5 Discussion and Conclusion

We introduced a novel SMB level generation approach using a diffusion model with the UNet architecture and Performer self-attention layers. Our model incorporates unique techniques, such as categorical data representation, reconstruction loss, and per-sprite temperature scaling, fostering high-quality and diverse SMB levels adhering to design principles.

Our evaluation explored the impact of beta scheduling schemes and per-sprite temperature adjustments on performance. Qualitatively, the generated levels effectively captured original design principles. Quantitatively, both evaluated versions exhibited varying levels of diversity, maintaining a balanced trade-off despite having slightly lower playability than the MarioGPT baseline. Refined temperature adjustments and further training may enhance performance, generating levels closely resembling actual levels while preserving high diversity and playability.

This research sets the stage for future work, including investigations of different noise schedules, optimization techniques, and self-attention mechanisms for efficient level generation. Manipulating per-sprite temperatures may enable generating levels favoring certain sprites, broadening the current random level generation scope. Additionally, assessing our model's performance in other 2D platformers or diverse game genres is worth further exploration.

In conclusion, our diffusion model provides a new perspective in the PCG domain, showcasing the potential of advanced deep learning techniques for SMB level generation. Capturing spatial and temporal relationships within SMB levels, our approach generates diverse, high-quality levels exhibiting both playability and visual fidelity, highlighting our substantial contribution to the field.

References

[1] Noor Shaker, Julian Togelius, and Mark J. Nelson. *Procedural Content Generation in Games*. Springer International Publishing, 1st edition, 2016.

[2] Jialin Liu, Sam Snodgrass, Ahmed Khalifa, Sebastian Risi, Georgios N Yannakakis, and Julian Togelius. Deep learning for procedural content generation. *Neural Computing and Applications*, 33(1):19–37, 2021.

[3] Jordan Pearson. Why artificial intelligence researchers love 'super mario bros.', Oct 2015.

[4] Lucas Ferreira, Leonardo Pereira, and Claudio Toledo. A multi-population genetic algorithm for procedural generation of levels for platform games. In *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation*, pages 45–46, 2014.

[5] Sam Snodgrass and Santiago Ontañón. A hierarchical approach to generating maps using markov chains. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 10, pages 59–65, 2014.

[6] Adam Summerville and Michael Mateas. Super mario as a string: Platformer level generation via lstms. *arXiv preprint arXiv:1603.00930*, 2016.

[7] Vanessa Volz, Jacob Schrum, Jialin Liu, Simon M Lucas, Adam Smith, and Sebastian Risi. Evolving mario levels in the latent space of a deep convolutional generative adversarial network. In *Proceedings of the genetic and evolutionary computation conference*, pages 221–228, 2018.

[8] Anurag Sarkar and Seth Cooper. Sequential segment-based level generation and blending using variational autoencoders. In *Proceedings of the 15th International Conference on the Foundations of Digital Games*, pages 1–9, 2020.

[9] Shyam Sudhakaran, Miguel González-Duque, Claire Glanois, Matthias Freiberger, Elias Najarro, and Sebastian Risi. Mariogpt: Open-ended text2level generation through large language models. *arXiv preprint arXiv:2302.05981*, 2023.

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[11] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[12] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[13] Adam James Summerville, Sam Snodgrass, Michael Mateas, and Santiago Ontanón. The vglc: The video game level corpus. *arXiv preprint arXiv:1606.07487*, 2016.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[15] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[17] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020.

[18] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[19] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.