To Infinity and Beyond: SHOW-1 and Showrunner Agents in Multi-Agent Simulations

Philipp Maas
Fable Studio

Frank Carey
Fable Studio

Chris Wheeler
Fable Studio

Edward Saatchi
Fable Studio

Pete Billington
Fable Studio

Jessica Yaffa Shamash
Fable Studio

Abstract

In this work we present our approach to generating high-quality episodic content for IP's (Intellectual Property) using large language models (LLMs), custom state-of-the art diffusion models and our multi-agent simulation for contextualization, story progression and behavioral control. Powerful LLMs such as GPT-4 were trained on a large corpus of TV show data which lets us believe that with the right guidance users will be able to rewrite entire seasons.*"That Is What Entertainment Will Look Like. Maybe people are still upset about the last season of Game of Thrones. Imagine if you could ask your A.I. to make a new ending that goes a different way and maybe even put yourself in there as a main character or something."*.[1]

[1]Brockman https://www.hollywoodreporter.com/business/digital/chatgpt-game-of-thrones-openai-greg-brockman-1235348099/amp/

1 Creative limitations of existing generative AI Systems

Current generative AI systems such as Stable Diffusion (Image Generator) and ChatGPT (Large Language Model) excel at short-term general tasks through prompt engineering. However, they do not provide contextual guidance or intentionality to either a user or an automated generative story system (showrunner[2]) as part of a long-term creative process which is often essential to producing high-quality creative works, especially in the context of existing IP's.

1.1 Living with uncertainty

Figure 1: Example Still from South Park AI Episode

By using a multi-agent[3] simulation as part of the process we can make use of data points such as a character's history, their goals and emotions, simulation events and localities to generate scenes and image assets more coherently and consistently aligned with the IP story world. The IP-based simulation also provides a clear, well known context to the user which allows them to judge the generated story more easily. Moreover, by allowing them to exert behavioral control over agents, observe their actions and engage in interactive conversations, the user's expectations and intentions are formed which we then funnel into a simple prompt to kick off the generation process.

Our simulation is sufficiently complex and non-deterministic to favor a positive disconfirmation. Amplification effects can help mitigate what we consider an undesired "slot machine" effect which we'll briefly touch on later. We are used to watching episodes passively and the timespan between input and "end of scene/episode" discourages immediate judgment by the user and as a result reduces their desire to "retry". This disproportionality of the user's minimal input prompt and the resulting high-quality long-form output in the form of a full episode is a key factor for positive disconfirmation.

While using and prompting a large language model as part of the process can introduce *"several challenges"*.[4] Some of them, like hallucinations, which introduce uncertainty or in more creative terms "unexpectedness", can be regarded as creative side-effects to influence the expected story outcome in positive ways. As long as the randomness introduced by hallucination does not lead to implausible plot or agent behavior and the system can recover, they act as happy-accidents[5], a term often used during the creative process, further enhancing the user experience.

[2] https://fablesimulation.com/blog/friends-ai-sitcom-simulation
[3] Sung Park https://arxiv.org/abs/2304.03442
[4] Li https://arxiv.org/abs/2303.17760
[5] Maas https://noproscenium.com/from-a-i-character-to-sundance-filmmaker-with-gpt-3-d4ab80c31b4e

1.2 The Issue of 'The Slot Machine Effect' in current Generative AI tools

The Slot Machine Effect refers to a scenario where the *generation of AI-produced content feels more like a random game of chance rather than a deliberate creative process*[6]. This is due to the often unpredictable and instantaneous nature of the generation process.

Current off-the-shelf generative AI systems do not support or encourage multiple creative evaluation steps in context of a long-term creative goal. Their interfaces generally feature various settings, such as sliders and input fields which increase the level control and variability. The final output however, is generated almost instantaneously by the press of a button. This instantaneous generation process results in immediate gratification providing a dopamine rush to the user. This reward mechanism would be generally helpful to sustain a multi-step creative process over long periods of time but current interfaces, the frequency of the reward and a lack of progression (stuck in an infinite loop) can lead to negative effects such as frustration, the intention-action gap[7] or a *loss of control over the creative process. The gap results from behavioral bias favoring immediate gratification*, which can be detrimental to long-term creative goals.

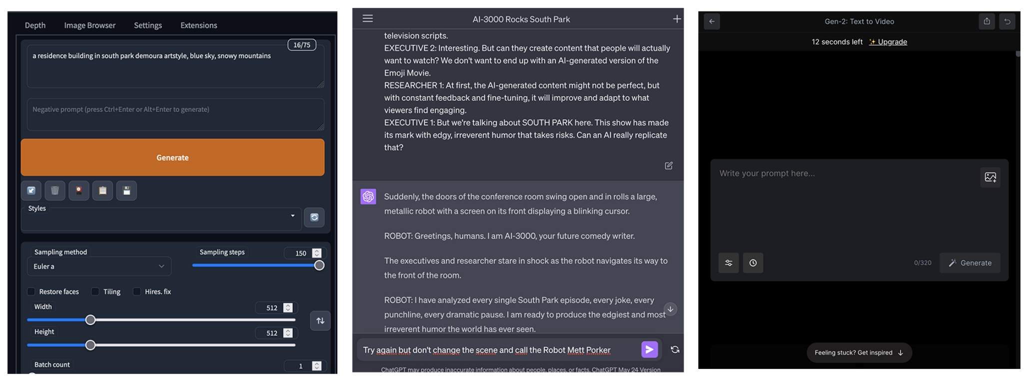

Figure 2: User Interface Comparison - Left to right: *Stable Diffusion Gradio, ChatGPT, Runway Gen-2*

While we do not directly solve these issues through interfaces, the contextualization of the process in a simulation and the above mentioned disproportionality and timespan between input and output help mitigate them. In addition we see opportunities in the simulation for in-character discriminators that participate in the creative evaluation process, such as an agent reflecting on the role they were assigned to or a scene they should perform in.

The multi-step "trial and error" process of the generative story system is not presented to the user, therefore it doesn't allow for intervention or judgment, avoiding the negative effects of immediate gratification through a user's "accept or reject" decisions. It does not matter to the user experience how often the AI system has to retry different prompt chains[8] as long as the generation process is not negatively perceived as idle time but integrated seamlessly with the simulation gameplay. The user only acts as the discriminator in the very end of the process after having watched the generated scene or episode. This is also an opportunity to utilize the concept of Reinforcement Learning through Human Feedback (RLHF) for improving the multi-step creative process and as a result the automatically generated episode.

1.3 Large Language Models

LLMs represent the forefront of natural language processing and machine learning research, demonstrating exceptional capabilities in understanding and generating human-like text. They are typically

[6]https://artificial.tech/slot-machine-effect-of-ai/
[7]https://thedecisionlab.com/reference-guide/psychology/intention-action-gap
[8]Yang https://arxiv.org/abs/2306.02224

built on Transformer-based architectures, a class of models that rely on self-attention[9] mechanisms. Transformers allow for efficient use of computational resources, enabling the training of significantly larger language models. GPT-4, for instance, comprises billions of parameters that are trained on extensive datasets, effectively encoding a substantial quantity of worldly knowledge in their weights.

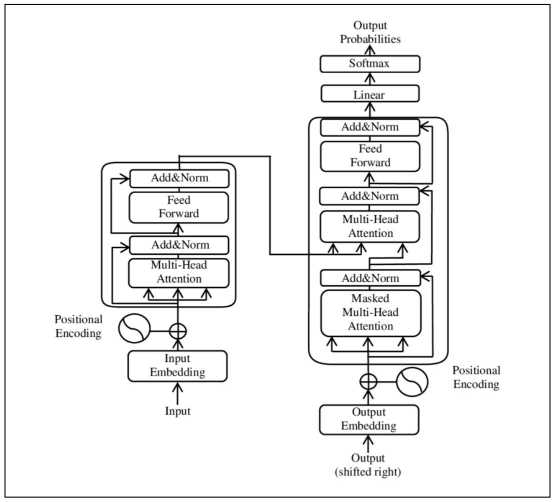

Figure 3: Diagram of the Transformer Architecture[10]

Central to the functioning of these LLMs is the concept of vector embeddings. These are mathematical representations of words or phrases in a high-dimensional space. These embeddings capture the semantic relationships between words, such that words with similar meanings are located close to each other in the embedding space. In the case of LLMs, each word in the model's vocabulary is initially represented as a dense vector, also known as an embedding. These vectors are adjusted during the training process, and their final values, or "embeddings", represent the learned relationships between words. During training, the model learns to predict the next word in a sentence by adjusting the embeddings and other parameters to minimize the difference between the predicted and actual words. The embeddings thus reflect the model's understanding of words and their context. Moreover, because Transformers can attend to any word in a sentence regardless of its position, the model can form a more comprehensive understanding of the meaning of a sentence. This is a significant advancement over older models that could only consider words in a limited window. The combination of vector embeddings and Transformer-based architectures in LLMs facilitates a deep and nuanced understanding of language, which is why these models can generate such high-quality, human-like text.

As was mentioned previously, transformer-based language models excel at short-term general tasks. They are regarded as fast-thinkers. [Kahneman][12]. Fast thinking pertains to instinctive, automatic, and often heuristic-based decision-making, while slow thinking involves deliberate, analytical, and effortful processes. LLMs generate responses swiftly based on patterns learned from training data, without the capacity for introspection or understanding the underlying logic behind their outputs. However, this also implies that LLMs lack the ability to deliberate, reason deeply, or learn from singular experiences[13] in the way that slow-thinking entities, such as humans, can. While these models have made remarkable strides in text generation tasks, their fast-thinking nature may limit their potential in tasks requiring deep comprehension or flexible reasoning. More recent approaches to imitate slow-thinking capabilities such as prompt-chaining (see Auto-GPT) showed promising results. Large language models seem powerful enough to act as their own discriminator in a multi-step

[9] Vaswani https://arxiv.org/abs/1706.03762

[10] Vaswani https://arxiv.org/abs/1706.03762

[11] https://techcommunity.microsoft.com/t5/azure-data-explorer-blog/azure-data-explorer-for-vector-similarity-search/ba-p/3819626

[12] Bubeck https://arxiv.org/abs/2303.12712

[13] Bubeck https://arxiv.org/abs/2303.12712

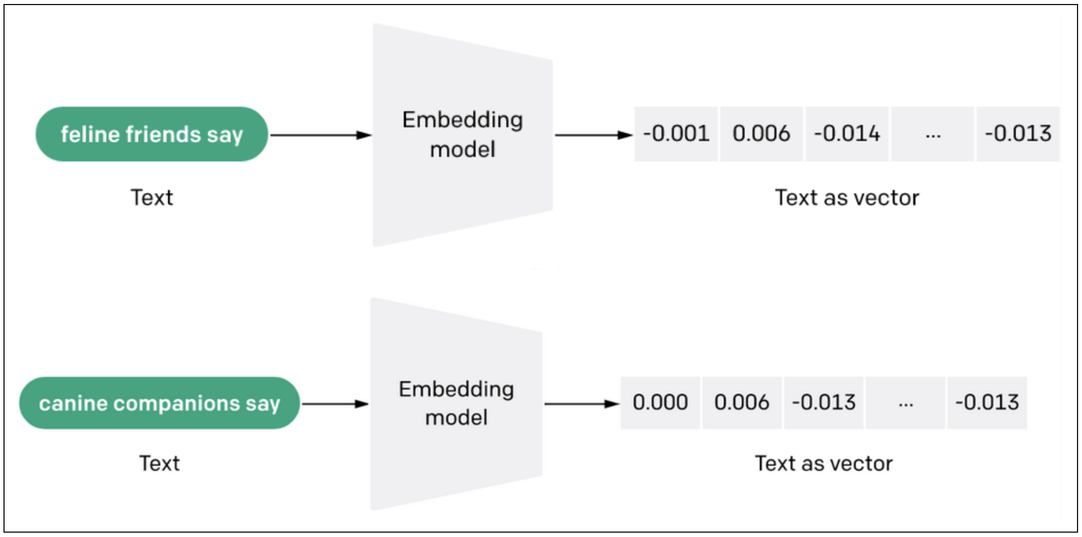

Figure 4: Example of Text Vector Embedding[11]

process. *This can dramatically improve the ability to reason in different contexts, such as solving math problems.*[14]

We make heavy use of GPT-4 to influence the agents in the simulation as well as generating the scenes for the south park episode. Since transcriptions of most of the south park episodes are part of GPT-4's training dataset, it already has a good understanding of the character's personalities, talking style as well as overall humor of the show, eliminating the need for a custom fine-tuned model.

However, we do imitate slow-thinking as part of a multi-step creative process. For this we use different prompt chains to compare and evaluate the events of different scenes and how they progress the overall story towards a satisfactory, IP-aligned result. Our attempt to generate episodes through prompt-chaining is due to the fact that story generation is a highly discontinuous task.[15] *These are tasks where the content generation cannot be done in a gradual or continuous way, but instead requires a certain "Eureka" idea that accounts for a discontinuous leap in the progress towards the solution of the task. The content generation involves discovering or inventing a new way of looking at or framing the problem, that enables the generation of the rest of the content. Examples of discontinuous tasks are solving a math problem that requires a novel or creative application of a formula, writing a joke or a riddle, coming up with a scientific hypothesis or a philosophical argument, or creating a new genre or style of writing.*

1.4 Diffusion Models

Diffusion models operate on the principle of gradually adding or removing random noise from data over time to generate or reconstruct an output. The image starts as random noise and, over many steps, gradually transforms into a coherent picture, or vice versa.

In order to train our custom diffusion models, we collected a comprehensive dataset comprising approximately 1200 characters and 600 background images from the TV show South Park. This dataset serves as the raw material from which our models learned the style of the show.

To train these models, we employ Dream Booth.[16] The result of this training phase is the creation of two specialized diffusion models.

The first model is dedicated to generating single characters set against a keyable background color. This facilitates the extraction of the generated character for subsequent processing and animation, allowing us to seamlessly integrate newly generated characters into a variety of scenes and settings. h

[14] Baker https://openai.com/research/improving-mathematical-reasoning-with-process-supervision
[15] Bubeck https://arxiv.org/abs/2303.12712
[16] Ruiz https://arxiv.org/abs/2208.12242

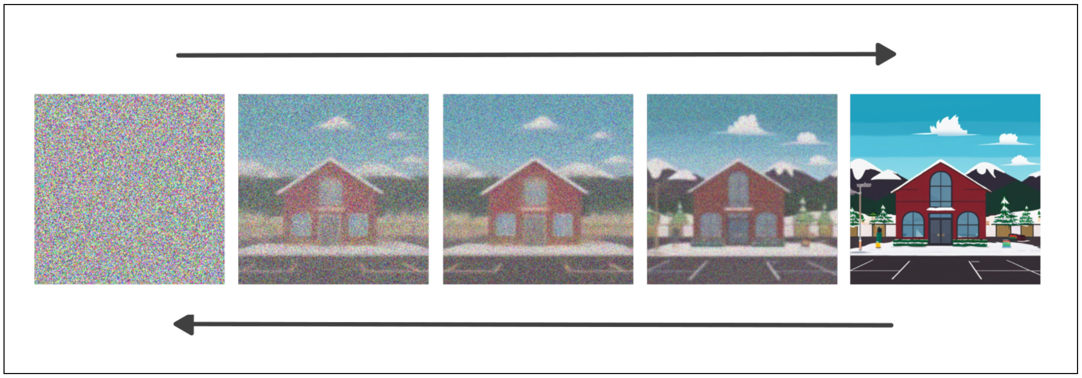

Figure 5: Stable Diffusion Model for South Park Backgrounds, prompt: *"a residence building in South park [demoura artstyle]"*

Figure 6: Example of generated South Park character

In addition, the character diffusion model allows the user to create a south park character based on their own looks via the image-to-image process of stable diffusion and then join the simulation as an equally participating agent. With the ability to clone their own voice, it's easy to imagine a fully realized autonomous character based on the user's characteristic looks, writing style and voice.

The second model is trained to generate clean backgrounds, with a particular focus on both exterior and interior environments. This model provides the 'stage' upon which our generated characters can interact, allowing for a wide range of potential scenes and scenarios to be created.

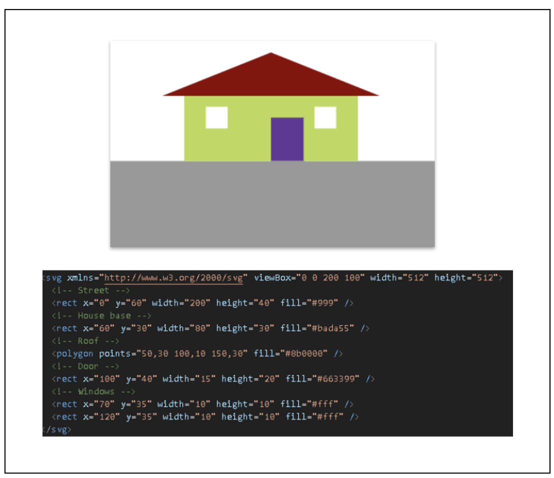

Figure 7: GPT-4 generated SVG image, prompt: *"Can you give me a svg drawing of a house on a street?"*

However, it's important to note that the images produced by these models are inherently limited in their resolution due to the pixel-based nature of the output. To circumvent this limitation, we post-process the generated images using an AI upscaling technique, specifically R-ESRGAN-4x+-Anime6B, which refines and enhances the image quality.

For future 2D interactive work, training custom transformer based models that are capable of generating vector-based output would have several advantages. Unlike pixel-based images, vector graphics do not lose quality when resized or zoomed, thus offering the potential for infinite resolution. This will enable us to generate images that retain their quality and detail regardless of the scale at which they are viewed. Furthermore, vector based shapes are already separated into individual parts, solving pixel-based post-processing issues with transparency and segmentation which complicate the integration of generated assets into procedural world building and animation systems.

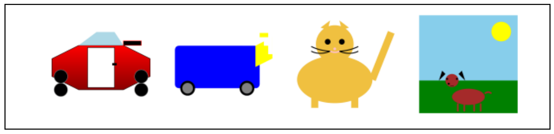

Figure 8: SVGs generated by GPT-4 for the classes automobile, truck, cat, dog.[17]

2 Episode Generation

We define an episode as a sequence of dialogue scenes in specific locations which add up to a total runtime of a regular 22 min south park episode.

In order to generate a full south park episode, we prompt the story system with a high level idea, usually in the form of a title, synopsis and major events we want to see happen over the course of 1 week in simulation time (=roughly 3 hours of play time).

From this, the story system automatically extrapolates up to 14 scenes by making use of simulation data as part of a prompt chain. The showrunner system takes care of casting the characters for each scene and how the story should progress through a plot pattern. Each scene is associated with a plot letter (e.g. A, B, C) which is then used by the showrunner to alternate between different character groups and follow their individual storylines over the course of an episode to keep the user engaged.

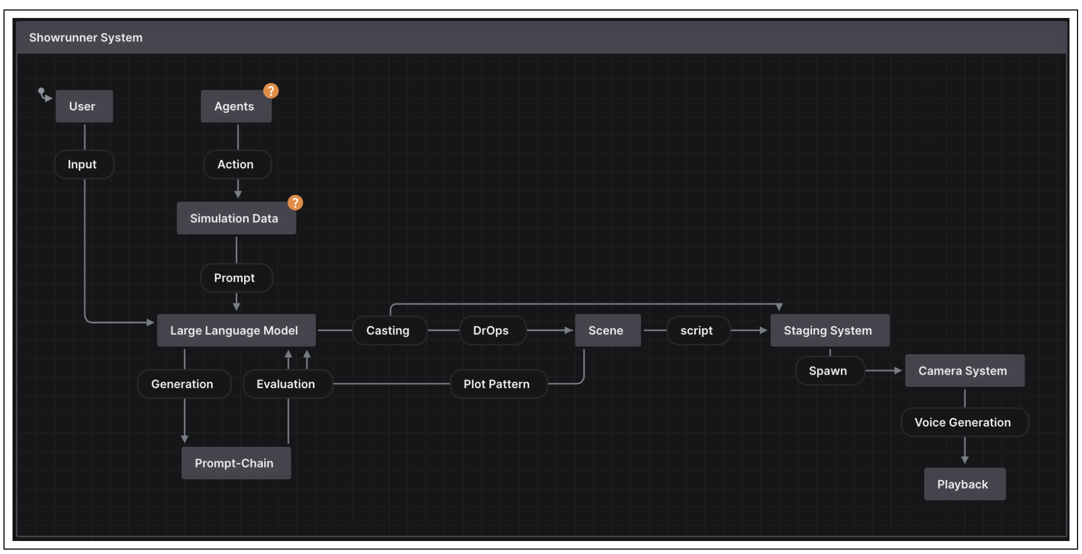

Figure 9: Diagram of Showrunner systems and prompt graph

In the end, each scene simply defines the location, cast and dialogue for each cast member. The scene is played back according to the plot pattern (e.g. ABABC) after the staging system and AI camera system went through initial setup. The voice of each character has been cloned in advance and voice clips are generated on the fly for every new dialogue line.

[17]https://arxiv.org/abs/2303.12712

2.1 Reducing Latency

In our experiments, generating a single scene can take a significant amount of time of up to one minute. Below is a response time comparison between GPT-3.5-turbo and GPT-4. Speed will increase in the short-term as models and service infrastructure get improved and other factors like artificial throttling due to high user demand will get removed.

Since we generate the episodes during gameplay, we have ways to hide most of the generation time in moments when the user is still interacting with the simulation or other user interfaces. Another way to reduce the time needed to generate a scene or episode is to use faster models such as GPT-3.5-turbo for specific prompts in the chain where the highest quality and accuracy is not so important.

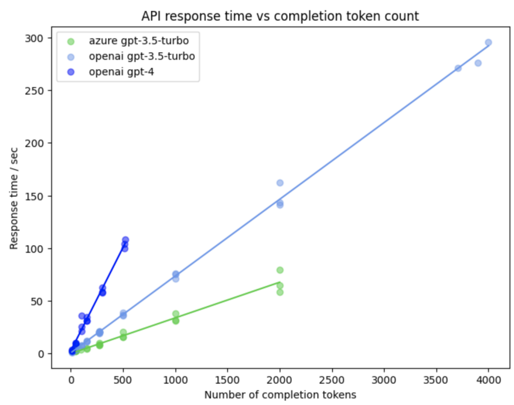

Figure 10: Speed comparison of GPT-3.5 vs. GPT-4[18]

During scene playback, we avoid any unwanted pauses between dialogue lines related to audio generation by using a simple buffering system which generates at least one voice clip in advance. See figure 11. This means while one character is delivering their voice clip, we already make the web request for the next voice clip, wait for it to generate, download the file and then wait for the current speaker to finish his dialogue before playback (delay). In this way the next dialogue line's voice clip is always delivered without any delay. Text generation and voice cloning services become increasingly fast and allow for highly adaptive and near-real time voice conversations.

2.2 Simulate creative thinking

As stated earlier, the data produced by the simulation acts as creative fuel to both, the user who is writing the initial prompt and the generative story system which is interacting with the LLM via prompt-chaining. Prompt-chaining[19] is a technique, which involves supplying the language model

[18] Pungas https://www.taivo.ai/$_{gpt-3-5-and-gpt-4-response-times/}$

[19] Wu https://arxiv.org/abs/2203.06566

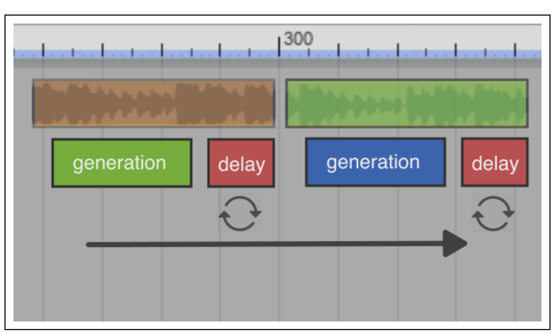

Figure 11: Diagram of zero-delay voice clip generation

with a sequence of related prompts to simulate a continuous thought process. Sometimes it can take on different roles in each step to act as the discriminator against the previous prompt and generated result.

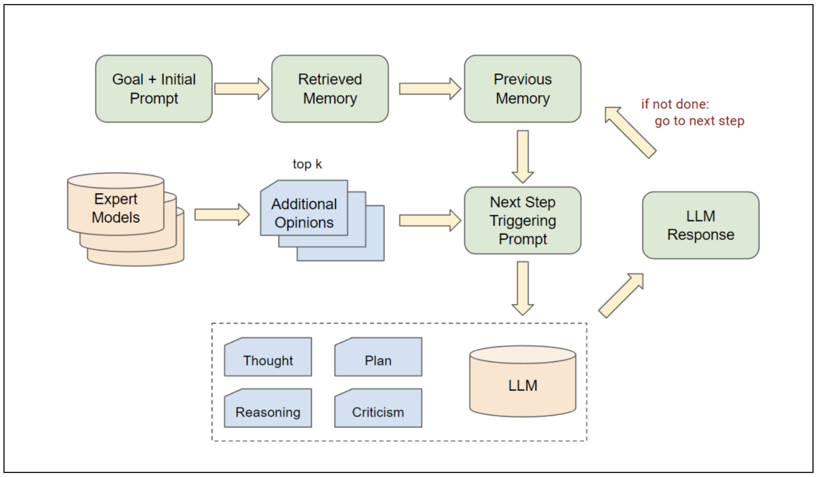

Figure 12: Example of a prompt chain from Auto-GPT[20]

In our case we try to mimic that of a discontinuous creative thought process. For example, the creation of 14 distinct South Park scenes could be managed by initially providing a broad prompt to outline the general narrative, followed by specific prompts detailing and evaluating each scene's cast, location, and key plot points. This mimics the process of human brainstorming, where ideas are built upon and refined in multiple often discontinuous steps. By leveraging the generative capabilities of LLMs in conjunction with the iterative refinement offered by prompt-chaining, we can effectively construct a dynamic, detailed, and engaging narrative.

In addition, we explore new concepts like plot patterns and dramatic operators (DrOps) to enhance the episode structure overall but also the connective tissue between each scene. Stylistic devices like reversals, foreshadowing, cliffhangers are difficult to evaluate as part of a prompt chain. A user without a writing background would have equal difficulty in judging these stylistic devices for their effectiveness and proper placement. We propose a procedural approach, injecting these show specific patterns and stylistic devices into the prompt chain programmatically as plot patterns and DrOps which can operate on the level of act structures, scene structures and individual dialogue lines. We are investigating future opportunities to extract what we call a dramatic fingerprint which is specific to each IP and format and train our custom SHOW-1 model with these data points. This dataset combined with overall human feedback could further align tone, style and entertainment value between the user and the specified IP while offering a highly adaptive and interactive story system as part of the on-going simulation.

2.3 Blank Page Problem

As mentioned above, one of the advantages of the simulation is that it avoids the blank page problem for both a user and a large language model by providing creative fuel[22]. Even experienced writers can sometimes feel overwhelmed when asked to come up with a title or story idea without any prior incubation of related material. The same could be said for LLMs. The simulation provides context and data points before starting the creative prompt chain.

[20]Yank, Auto-GPT for Online Decision Making: Benchmarks and Additional Opinions https://arxiv.org/pdf/2306.02224.pdf

[21]Drhlík, https://pdrhlik.github.io/southparktalk-whyr2018/

[22]https://www.trytriggers.com/blog-posts/overcoming-the-barrier-of-the-blank-page

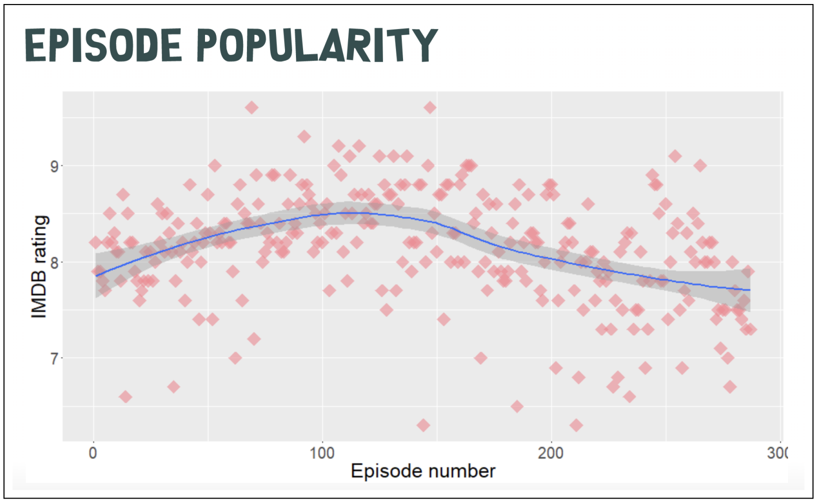

Figure 13: Diagram of South Park Episode ratings from IMDB[21]

Figure 14: Example UI of Showrunner prompt input

2.4 Who is driving the story

The story generation process in our approach is a shared responsibility between the simulation, the user, and GPT-4. Each has strengths and weaknesses and a unique role to play depending on how much we want to involve them in the overall creative process. Their contributions can have different weights. While the simulation usually provides the foundational IP-based context, character histories, emotions, events, and localities that seed the initial creative process. The user introduces their intentionality, exerts behavioral control over the agents and provides the initial prompts that kick off the generative process. The user also serves as the final discriminator, evaluating the generated story content at the end of the process. GPT-4, on the other hand, serves as the main generative engine, creating and extrapolating the scenes and dialogue based on the prompts it receives from both the user and the simulation. It's a symbiotic process where the strengths of each participant contribute to a coherent, engaging story. Importantly, our multi-step approach in the form of a prompt-chain also provides checks and balances, mitigating the potential for unwanted randomness and allowing for more consistent alignment with the IP story world.

2.5 SHOW-1 and Intentionality

The formular (creative characteristics) and format (technical characteristics) of a show are often a function of real-world limitations and production processes. They usually don't change, even over the course of many seasons (South Park currently has 26 seasons and 325 episodes).[23]

A single dramatic fingerprint of a show, which is used to train the proposed SHOW-1 model, can be regarded as a highly variable template or "formula" for a procedural generator that produces South Park-like episodes.

To train a model such as SHOW-1 we need to gather a sufficient amount of data points in relation to each other that characterize a show. A TV show does not just come into existence and is made up of the final dialogue lines and set descriptions as seen by the audience. Existing datasets on which current LLM's are trained on only consist of the final screenplay which has the cast, dialogue lines and sometimes a short scene header. A lot of information is missing, such as timing, emotional states, themes, contexts discussed in the writer's room and detailed directorial notes to give a few examples. The development and refinement of characters is also part of this on-going process. Fictional characters have personalities, backstories and daily routines which help authors to sculpt not only scenes but the arcs of whole seasons. Even during a show characters keep evolving based on audience feedback or changes in creative direction. With the Simulation, we can gather data continuously from both the user's input and the simulated agents. Over time, as episodes are created, refined and rated by the user we can start to train a show specific model and deploy it as a checkpoint which allows the user to continue to refine and iterate on either their own original show or alternatively push an already existing show such as south park into directions previously not conceived by the original show runners and IP holders. To illustrate this, we imagine a user generating multiple south park episodes in which Cartman, one of the main characters and known for his hot headedness, slowly changes to be shy and naive while the life of other characters such as Butters could be tuned to follow a much more dominant and aggressive path. Over time, this feedback loop of interacting with and fine-tuning the SHOW-1 model can lead to new interpretations of existing shows but more excitingly to new original shows based on the user's intention. One of the challenges in order to make this feedback loop engaging and satisfying is the frequency at which a model can be trained. A model which is fed by real-time simulation data and user input should not feel static or require expensive resources to adapt. Otherwise the output it generates can feel static and unresponsive as well.

When a generative system is not limited in its ability to swiftly produce high amounts of content and there is no limit for the user to consume such content immediately and potentially simultaneously, the *10,000 Bowls of Oatmeal*[24] problem can become an issue. Everything starts to look and feel the same or even worse, the user starts to recognize a pattern which in turn reduces their engagement as they expect newly generated episodes to be like the ones before it, without any surprises.

This is quite different from a predictable plot which in combination with the above mentioned "positive hallucinations" or happy accidents of a complex generative system can be a good thing. Surprising the user by balancing and changing the phases of certainty vs. uncertainty helps increase their overall engagement. If they would not expect or predict anything, they could also not get pleasantly surprised.

With our work we aim for perceptual uniqueness. The OatMeal problem of procedural generators is mitigated by making use of an on-going simulation (a hidden generator) and the long-form content of 22 min episodes which are only generated every 3h. This way the user generally does not consume a high quantity of content simultaneously or in a very short amount of time. This artificial scarcity, natural game play limits and simulation time help.

Another factor that keeps audiences engaged while watching a show and what makes episodes unique is intentionality from the authors. A satirical moral premise, twisted social commentary, recent world events or cameos by celebrities are major elements for South Park. Other show types, for example sitcoms, usually progress mainly through changes in relationship (some of which are never fulfilled), keeping the audience hooked despite following the same format and formula.

Intentionality from the user to generate a high-quality episode is another area of internal research. Even users without a background in dramatic writing should be able to come up with stories, themes

[23]https://en.wikipedia.org/wiki/South$_Park$

[24]Compton, Procedural Storytelling in Game Design

or major dramatic questions they want to see played out within the simulation. To support this, the showrunner system could guide the user by sharing its own creative thought process and make encouraging suggestions or prompting the user by asking the right questions. A sort of reversed prompt engineering where the user is answering questions.

One of the remaining unanswered questions in the context of intentionality is how much entertainment value (or overall creative value) is directly attributed to the creative personas of living authors and directors. Big names usually drive ticket sales but the creative credit the audience gives to the work while consuming it seems different. Watching a Disney movie certainly carries with it a sense of creative quality, regardless of famous voice actors, as a result of brand attachment and its history.

AI generated content is generally perceived as lower quality and the fact that it can get generated in abundance further decreases its value. How much this perception would change if Disney were to openly pride themselves on having produced a fully AI generated movie is hard to say. What if Steven Spielberg, single handedly generated an AI movie? Our assumption is that the perceived value of AI generated content would certainly increase.

A new interesting approach to replicate this could be the embodiment of creative AI models such as SHOW-1 to allow them to build a persona outside their simulated world and build relationships via social media[25] or real world events with their audience.[26] As long as an AI model is perceived as a black box and does not share their creative process and reasoning in a human and accessible way, as is the case for living writers and directors, it's unlikely to get credit with real creative values. However, for now this is a more philosophical question in the context of AGI.

3 Conclusion

Our approach of using multi-agent simulation and large language models for generating high-quality episodic content provides a novel and effective solution to many of the limitations of current AI systems in creative storytelling. By integrating the strengths of the simulation, the user, and the AI model, we provide a rich, interactive, and engaging storytelling experience that is consistently aligned with the IP story world. Our method also mitigates issues such as the 'slot machine effect', 'the oatmeal problem' and 'blank page problem' that plague conventional generative AI systems. As we continue to refine this approach, we are confident that we can further enhance the quality of the generated content, the user experience, and the creative potential of generative AI systems in storytelling.

Acknowledgements We are grateful to Lewis Hackett for his help and expertise in training the custom Stable Diffusion Models.

References

[1] Brockman https://www.hollywoodreporter.com/business/digital/chatgpt-game-of-thrones-openai-greg-brockman-1235348099/amp/

[2] Maas https://fablesimulation.com/blog/friends-ai-sitcom-simulation

[3] Sung Park https://arxiv.org/abs/2304.03442

[4] Li https://arxiv.org/abs/2303.17760

[5] Maas https://noproscenium.com/from-a-i-character-to-sundance-filmmaker-with-gpt-3-d4ab80c31b4e

[6] https://artificial.tech/slot-machine-effect-of-ai/

[7] https://thedecisionlab.com/reference-guide/psychology/intention-action-gap

[8] Yang https://arxiv.org/abs/2306.02224

[9] Vaswani https://arxiv.org/abs/1706.03762

[10] Vaswani https://arxiv.org/abs/1706.03762

[25]Virtual Beings https://www.youtube.com/watch?v=FSq-mheA7Ds, https://www.youtube.com/watch?v=IROZSq–MQE

[26]Collaborating with AI at Sundance https://www.fable-studio.com/behind-the-scenes/ai-collaboration

[11] https://techcommunity.microsoft.com/t5/azure-data-explorer-blog/azure-data-explorer-for-vector-similarity-search/ba-p/3819626

[12] Bubeck https://arxiv.org/abs/2303.12712

[13] Bubeck https://arxiv.org/abs/2303.12712

[14] Baker https://openai.com/research/improving-mathematical-reasoning-with-process-supervision

[15] Bubeck https://arxiv.org/abs/2303.12712

[16] Ruiz https://arxiv.org/abs/2208.12242

[17] https://arxiv.org/abs/2303.12712

[18] Pungas https://www.taivo.ai/$_{gpt-3-5-and-gpt-4-response-times/}$

[19] Wu https://arxiv.org/abs/2203.06566

[20] Yank, Auto-GPT for Online Decision Making: Benchmarks and Additional Opinions https://arxiv.org/pdf/2306.02224.pdf [21] Drhlík, https://pdrhlik.github.io/southparktalk-whyr2018

[22] https://www.trytriggers.com/blog-posts/overcoming-the-barrier-of-the-blank-page

[23] https://en.wikipedia.org/wiki/South$_{P}ark$

[24] Compton, Procedural Storytelling in Game Design

[25] Virtual Beings https://www.youtube.com/watch?v=FSq-mheA7Ds, https://www.youtube.com/watch?v=IROZSq–MQE

[26] Collaborating with AI at Sundance https://www.fable-studio.com/behind-the-scenes/ai-collaboration