DreamFactory: Pioneering Multi-Scene Long Video Generation with a Multi-Agent Framework

Anonymous EMNLP submission

Abstract

Current video generation models excel at creating short, realistic clips, but struggle with longer, multi-scene videos. We introduce DreamFactory, an LLM-based framework that tackles this challenge. DreamFactory leverages multi-agent collaboration principles and a Key Frames Iteration Design Method to ensure consistency and style across long videos. It utilizes Chain of Thought (COT) to address uncertainties inherent in large language models. DreamFactory generates long, stylistically coherent, and complex videos. Evaluating these long-form videos presents a challenge. We propose novel metrics such as Cross-Scene Face Distance Score and Cross-Scene Style Consistency Score. To further research in this area, we contribute the Multi-Scene Videos Dataset containing over 150 human-rated videos. DreamFactory paves the way for utilizing multi-agent systems in video generation. We will make our framework and datasets public after paper acceptance.

1 Introduction

Video, integrating both visual and auditory modalities—the most direct sensory pathways through which humans perceive and comprehend the world—effectively conveys information with compelling persuasiveness and influence, progressively becoming a powerful tool and medium for communication [(Tang and Isaacs, 1992), (Owen and Wildman, 1992), (Armes, 2006), (Harris, 2016), (Merkt et al., 2011)]. Traditional video production is an arduous and time-intensive process, particularly for capturing elusive real-life scenes. Owing to the rapid advancements in deep learning, AI-driven video generation techniques now facilitate the acquisition of high-quality images and video segments with ease [(pika), (Blattmann et al., 2023a), (openai, a), (Blattmann et al., 2023b), (runway), (Gu et al., 2023)]. However, crafting practical, multi-scene videos that meet real-world needs remains a formidable challenge. This includes ensuring consistency in character portrayal, stylistic coherence, and background across different scenes, proficiently maneuvering professional linguistic tools, and managing complex production steps beyond merely assembling brief video clips generated by current technologies. Therefore, there is an urgent need within the field of video generation for a model capable of directly producing long-duration, high-quality videos with high consistency, thus enabling AI-generated video to gain widespread acceptance and become a premier producer of content for human culture and entertainment.

At the current stage, substantial advancements in the video domain utilize diffusion-based generative models, achieving excellent visual outcomes [(Blattmann et al., 2023a), (runway), (openai, a)]. Nonetheless, due to the intrinsic characteristics of diffusion models, the videos produced are typically short segments, usually limited to four seconds. For generating longer videos, models like LSTM and GANs are employed (Gupta et al., 2022), however, these models struggle to meet the demands for high image quality and are restricted to synthesizing videos of lower resolution. These state-of-the-art approaches attempt to use a single model to address all sub-challenges of video generation end-to-end, encompassing attractive scriptwriting, character definition, and artistic shot design. However, these tasks are typically collaborative and not the sole responsibility of a single model.

In addressing complex tasks and challenges in problem-solving and coding, researchers have begun utilizing LLM multi-agent collaborative techniques, modeled on human cooperative behaviors, and have observed numerous potent agents. With the integration of large models that include visual capabilities, multi-agent collaborative technologies have now developed an AI workflow capable of tackling challenges in the image and video domain.

In this paper, we introduce multi-agent collabora-

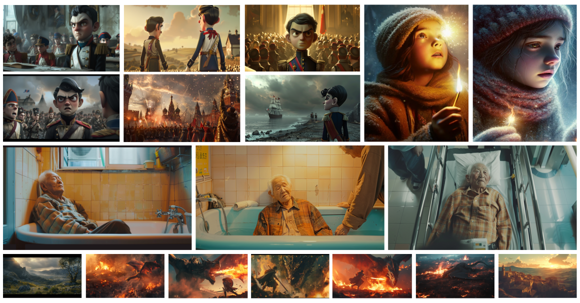

Figure 1: Keyframe data produced by **DreamFactory**. It can be seen that the character's facial features, visual style, and even clothing are consistent.

tive techniques to the domain of video generation, developing a multi-scene long video generation framework named **DreamFactory**, which simulates an AI virtual film production team. Agents based on LLMs assume roles akin to directors, art directors, screenwriters, and artists, collaboratively engaging in scriptwriting, storyboard creation, character design, keyframe development, and video synthesis. We define the concept of keyframe in the long video generation field to maintain consistency across video segments. In **DreamFactory**, we draw on the successful CoT concept from the multi-agent reasoning process to devise a keyframe iteration method specific to video. To address the drift phenomenon in large language models, a Monitor role is introduced to ensure consistency between different frames. **DreamFactory** also establishes an integrated image vector database to maintain the stability of the creative process. Based on the algorithms discussed, DreamFactory can automate the production of multi-scene videos of unrestricted length with consistent image continuity.

To evaluate our framework, we employed state-of-the-art video generation models as tools, measuring video generation performance on the UTF-101 and HMDB51 datasets. Furthermore, given the novelty of our task, with few prior ventures into this area, we compared long videos generated by our framework against those produced using the original tools. We found that our model significantly outperformed the existing native models regarding evaluation mechanisms. Finally, we collected AI-generated short videos currently available on the internet and assessed them using mechanisms such as the Inception Score, alongside evaluations conducted by human judges. Our findings indicate that our videos surpass the average quality of those produced manually. Some examples generated by the framework are shown in Figure 1.

2 Related work

LLM-based Agents. In recent years, the capabilities of large language models have been continually enhanced, exemplified by advancements such as GPT-4 (openai, b), Claude-3 (Claude), and LLama-2 (meta), among others. Subsequently, exploration into enhancing the abilities of these large language models has emerged, introducing methodologies such as CoT (Wei et al., 2022), ToT (Yao et al., 2024), ReACT (Yao et al., 2022), Reflexion (Shinn et al., 2024), and various other approaches to facilitate iterative output and correction cycles. Within this context, the notion of Multi-agents has surfaced, with early research efforts including notable works such as Camel (Li et al., 2024), Voyager (Wang et al., 2023a), MetaGPT (Hong et al., 2023), ChatDev (Qian et al., 2023), and Auto-GPT (Yang et al., 2023). Recently, powerful Multi-agents frameworks have proliferated across diverse domains, with prominent instances in fields such as coding, including notable contributions such as CodeAgent (Tang et al., 2024), CodeAct (Wang et al., 2024), and Codepori (Rasheed et al., 2024). Utilitarian tools such as Toolformer (Schick et al., 2024), HuggingGPT (Shen et al., 2024), Tool-llm (Qin et al., 2023), and WebGPT (Nakano et al., 2021) have also been employed. Other noteworthy endeavors encompass projects like WebArena (Zhou et al., 2023), RET-LLM (Modarressi et al., 2023), and OpenAGI (Ge et al., 2024), each contributing to the advancement and proliferation of Multi-agents paradigms.

Video synthesis. In the field of video generation, traditional methods primarily utilize Generative Adversarial Networks (GANs) for video creation, as demonstrated in the works of Tim Brooks et al. (Brooks et al., 2022) and the foundational contributions of Ian Goodfellow et al. (Goodfellow et al., 2014) However, in recent years, a significant shift has occurred towards leveraging the potent capabilities of diffusion processes, with pioneering research conducted by Jascha et al. (Esser et al., 2023), and Song et al. (Song et al., 2020). The forefront of this evolution is marked by the development of Latent Video Diffusion Models. This approach is exemplified in the seminal efforts of Andreas Blattmann et al. (Blattmann et al., 2023b), Gu et al. (Gu et al., 2023),Guo et al. (Guo et al., 2023), He et al. (He et al., 2022) and Wang et al. (Wang et al., 2023b). Currently, the most formidable advancements in this area are four main models: Pika (pika), Stable Video (Blattmann et al., 2023a), Runway (runway), and Sora (openai, a).

3 DreamFactory

Our DreamFactory framework utilizes multiple large language models (LLMs) to form a simulated animation company, taking on roles such as CEO, Director, and Creator. Given a story, they collaborate and create a video through social interaction and cooperation. This framework allows LLMs to simulate the real world by using small video generation models as tools to accomplish a massive task. This section details the methodology behind our innovative DreamFactory framework. We first describe the defined role cards in Section 3.1 and discuss the pipeline in Section 3.2. Finally, we will discuss the keyframe iteration design method.

3.1 Role Definition

In the architecture of our simulation animation company **DreamFactory**, the following roles are included: CEO, movie director, film producer, Screenwriter, Filmmaker, and Reviewer.Within the DreamFactory framework, they function similarly to their real-world counterparts, taking on roles such as determining the movie's style, writing scripts, and drawing.

The definition prompts for their roles primarily consist of three main parts: **Job**, **Task** and **Requirements**. For instance, the definition prompt for a movie's creator would include the following sentences: (a) You are the **Movie Art Director**. Now, we are both working at Dream Factory,... (b) Your job is to **generate a picture according to the scenery** given by the director...and (c) you must **obey the real-world rules**, like color unchanged... For tasks such as plot discussions, we also limit their discussions to not exceed a specific number of rounds (depending on the user's settings and the company's size definition). We have included the following prompt to ensure this: "You give me your thought and story, and we should brainstorm and critique each other's idea. After discussing more than 5 ideas, any of us must actively terminate the discussion by picking up the best style and replying with a single word <INFO>, followed by our latest style decision, e.g., cartoon style."

In Figure 3, panels (a) and (b) feature schematic illustrations of a character being defined and initiating role play. The complete architecture of the entire company is fully introduced in Figure 8. For each, we defined a role card, which contains: 1) The role name is put on the left-upper corner of each card; 2) The phases of the role involved are put on the right-upper corner of each card; 3) On each role card, we show the role-involved conversation and collaborative roles; 4) We show the intermediate output of the role on the right-hand side of the card; and 5) Finally, we put the corresponding files or content out of conversations on the bottom of the card.

3.2 DreamFactory Framework pipeline

In this section, we introduce the specific pipeline of DreamFactory. **Figure 2** illustrates the main phases and indicates which agents engage in conversations. Before delving into our entire pipeline, it's essential to first outline its fundamental components: phases and conversations. As depicted in **Figure 3** (c, a phase represents a complete stage that takes some textual or pictorial content as input. Agents, composed of GPT, engage in roleplay, discussion, and collaboration for processing, ultimately yielding some output. A conversation is a basic unit of a phase, with typically more than one round of conversation encompassed within a phase. After a fixed number of conversations, a phase is approaching its conclusion, at which point DreamFactory will save certain interim conclusions generated within this phase that we wish to retain. For instance, in the Phase style decision, the final conclusion will be preserved. Furthermore, during subsequent phases, DreamFactory will provide the necessary precedents, such as invoking previous styles and scripts when designing keyframes later on.

Recently, large language models were found to have their capabilities limited by finite reasoning abilities, akin to how overly complex situations in real life can lead to carelessness and confusion. Therefore, the main idea of this framework, in the video domain, is to decompose the creation of long videos into specific stages, allowing specific large models to play designated roles and leverage their powerful capabilities in analyzing specific problems. Like a real-life film production company, DreamFactory adopts a classic workflow, starting with scriptwriting followed by drawing. Overall, the framework encompasses six primary stages: **Task Definition**, **Style Decision**, **Story Prompting**, **Script Design**, and **Key-frame Design** . The

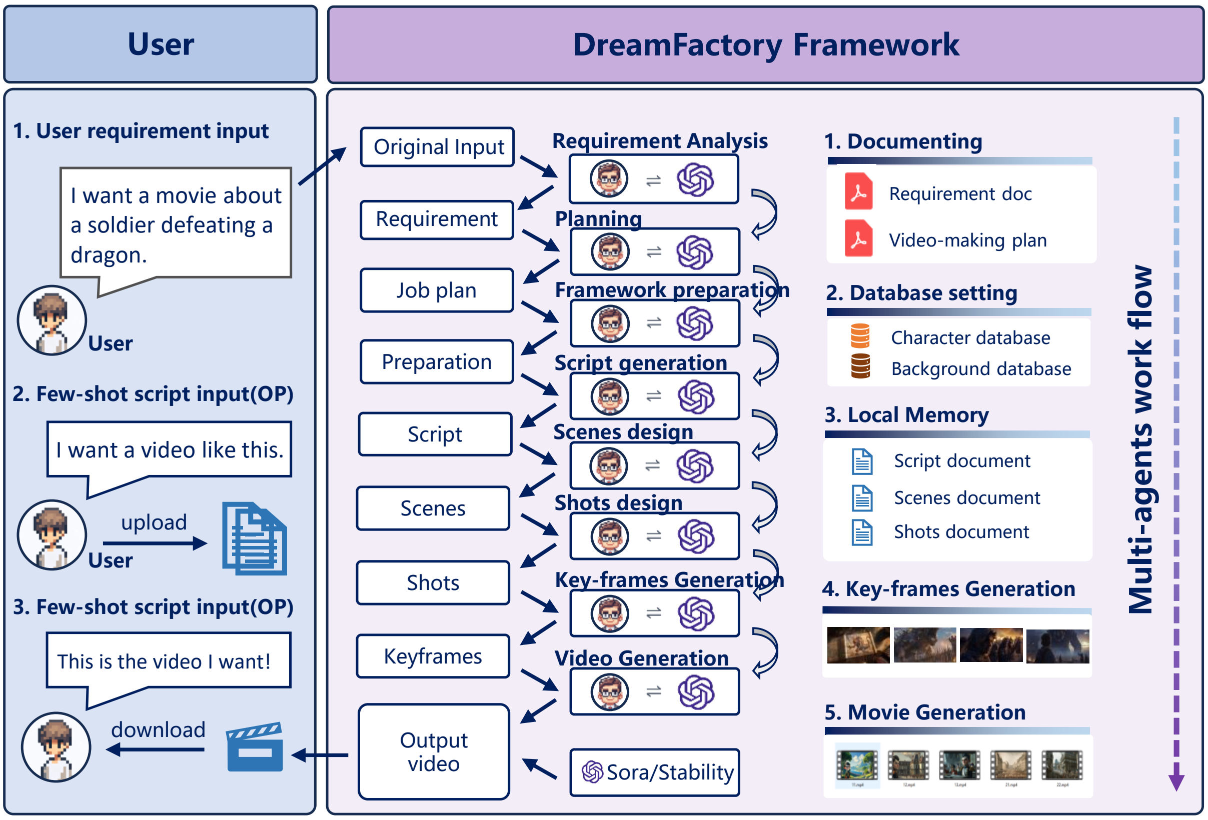

Figure 2: An overview of the DreamFactory framework. The framework transposes the entire filmmaking process into AI, forming an AI-driven video production team.

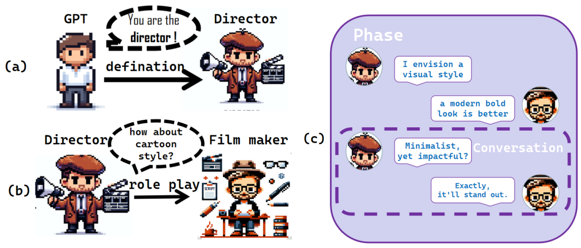

Figure 3: The Figure demonstrates how GPT begins role-playing as a director and commences communication with other GPTs as a director would.

specific method for the final stage, keyframe iterative design, will be introduced in the following section; it is used to maintain the consistency and continuity of images generated at various stages. In the first four phases, our roles are conversational.

In each phase, every agent shares a "phase prompt" that includes the following key points: our roles, our tasks, the conclusions we aim to draw, the form of our discussion, and some other requirements. Following this, each agent is further informed by its unique prompt about its role definition, as discussed in section 3.1. We can refer to the notation in Guohao Li's article[1] to define the collaboration process of agents within DreamFactory. We refer to the assistant system prompt/message by Pa and that of the user by Pu. The system messages are passed to the agents before the conversations start. Let F1 and F2 denote two large-scale autoregressive language models. When the system message is passed to those models respectively, we can get $A \leftarrow F_1^{P_A}$, $U \leftarrow F_2^{P_U}$ which are referred to as the assistant and user agents respectively. In continuation, we assume that the text provided by the user (instructor) at each instance is denoted as **It**, and the response given by the assistant is denoted as **At**. The Output at time step **t** alternating conversations between the two can be represented as: $\mathcal{O}_t = ((I_1, A_1), (I_2, A_2), \ldots, (I_t, A_t))$.

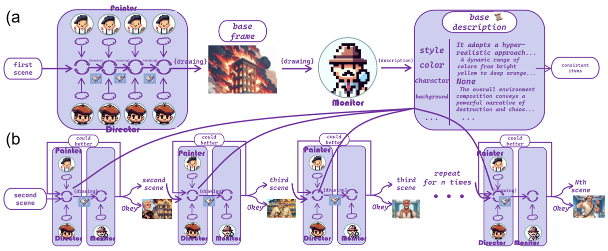

Figure 4: An overview of the keyframe iterative design.

Following the five critical phases mentioned above, five significant outputs will be achieved. In the prompt, each phase's output O_t is required to follow <INFO> for summarization, which also allows us to systematically obtain and preserve, forming the Local memory information of the `DreamFactory` framework. This is also one of the primary purposes of proposing this framework, maintaining the consistency of critical information. Finally, after generating the tasks, styles, stories, scripts, and keyframe images, a long video with consistent style is obtained.

3.3 Keyframe Iteration Design

During the generation of long videos, the most challenging problem to address is that a video comprises a long sequence of image collections. Therefore, when generating, the model needs to maintain a long-term, consistent memory to ensure that each frame produced by the model coherently composes a consistent video. This type of memory includes two kinds: **short-term memory knowledge** and **long-term memory system**.

short-term memory knowledge is embedded within videos of a fixed scene. Between adjacent frames, the animation in each frame should be connected, the characters should be unified, and there should be no significant changes in color, style, etc. As of now, the latest video models perform very well in terms of short-term memory. Nonetheless, we have still added a Monitor to supervise whether our video model is performing sufficiently well. As illustrated in Figure 4, there is a review process after the generation of each frame. Therefore, to maintain short-term consistency, the supervisory mechanism we introduced has addressed this issue.

long-term memory system, however, pose a challenge that troubles most current models and represents the most pressing issue in video generation today. Particularly, within a GPT-based fully automated multi-agent framework, the inherent randomness and drift phenomena of large language models make this problem difficult to tackle. Long-term memory implies that across scene transitions, the model should be able to maintain the consistency of the drawing style, character continuity, and narrative flow. To uphold long-term memory, we have introduced the Keyframe Iteration Design method, which transforms long-term memory into short-term memory by guiding the generation of consecutive, consistent images, iterating and generating forward with each step. **Figure 4** demonstrates the process of each iteration.

Keyframe Iteration Design Method leverages the inferential capabilities of large language models to transform long-term memory into iterations of short-term memory to ensure consistency. The first frame of the image is the beginning of the entire video and establishes essential information such as the style, painting technique, characters, and background for the entire long video. Therefore, we refer to the first frame as the Base. At the beginning, we will generate a painter P, a director D and a monitor M, represented by $P \leftarrow F_1^{P_P}$, $D \leftarrow F_2^{P_D}$, $M \leftarrow F_3^{P_M}$,these models played by visual large language models, will engage in a cyclical process of generation and discussion until they produce a crucial frame, which is the first keyframe, referred to as the **Base Frame**. At this point, the Monitor D, composed of a visual large language model as well, will conduct a thorough analysis to extract information, detailed description of features such as style, background, and character traits that should be preserved for an extended period. This results in the **Base Description**, note as B_D. $S1$ represents the script for the first frame. We have $\mathcal{O}_t = Gen\left(\ p_t, d_t, S_1\ \right)$, where $B_D \leftarrow M(O_t)$.

In subsequent generations, when iterating the keyframe for moment t, we will use the previously input S_t as the description of the scene. To maintain continuity in the context of adjacent scenes, we will employ the nurtured method to generate the description for the moment $t-1$, which we also refer to as the contextual environment denoted as $C_t - 1$. At the same time, to maintain long-distance memory, B_D will also serve as an input. By referencing the basic features of the previous frame and the Base features, it can ensure that the necessary information is essentially grasped in the next iteration, enabling the drawing of continuous keyframes with the same style, consistent characters, and uniform background. We have $\mathcal{O}_t = Gen\left(\ p_t, d_t, S_t, C_{t-1}\ \right)$.

Upon the previous generation of keyframes, we can obtain the contextual environment and proceed with the next round of generation. We have $C_t = M(O_t)$, $p_{t+1} = P(S_t, C_t)$, $d_{t+1} = D(S_t, C_t, p_{t+1})$. Ultimately, we achieve the generation of the keyframes for the moment $t+1$.

In practical application, controlling the details of characters proves to be the most challenging aspect. Therefore, under our carefully modified prompts, with increased emphasis on parts that performed poorly in multiple experiments, the Keyframe Iteration Method can now generate a very consistent and practically valuable series of images.

4 Experiments

4.1 Traditional Video Quality Evaluati1on

Evaluati1on Metrics - To validate the continuity of the keyframes and the quality of the videos produced by the framework, we embedded various tool models (such as Runway, Diffusion, GPT) within the architecture to assess the quality of videos generated by different tools. In our experiments, we principally employed the following evaluation metrics: **(1)** Fréchet Inception Distance (FID) score: measures the similarity between generated images and real images. **(2)** Inception Score (IS): gauges the quality and diversity of generated images. **(3)** CLIP Score: evaluates the textual description accuracy of generated images. **(4)** Fréchet Video Distance (FVD) score: extension of the FID for videos, comparing the features distribution of real videos versus synthesized ones based on Fréchet distance and **(5)** Kernel Video Distance (KVD): utilizes kernel function to compare the features distribution of real videos versus synthesized ones.

Our dataset, during the Regular phase, comprised conventional prompts consisting of 70 keywords and brief sentences randomly selected by experimental personnel from the COCO dataset. This was utilized to evaluate the generated image quality of the fundamental tool models and the degree of alignment between the images and the text. For the Script phase, scripts pertaining to 70 randomly extracted tasks from our provided dataset were employed during the script-filling stage. This guided the model generation based on the relevant plot to assess the function of the "Animation Department" within the `DreamFactory` framework. The `DreamFactory` label denotes the keyframe images produced by the framework that corresponds to the Script.

Output Quality Statistics - The images generated using models such as DALL·E and Diffusion are of high quality and have reached the state-of-the-art level in various indices. To quantitatively analyze the quality of the generated images, we input the images corresponding to the original prompts into GPT to get the GPT-Script and then used original prompts or the GPT-Script as prompts to generate 1400 images, from which we calculated FID, IS, and CLIP Score. As for FVD and KVD, we selected 100 samples from our multi-scene video dataset and manually extracted 10 keyframes for each one, Which can be used to generate multi-scale videos.

Data in Table 1 indicates that the quality of images generated using scripts is on average more refined than those produced using everyday prompt words. This may be attributable to the extent to which GPT acts as a prompt, and contemporary models are generally adept at processing longer prompts. However, within the `DreamFactory` framework, the application of keyframe iterative design, in conjunction with storyboard creation, detailed descriptions of characters, settings, lighting, and style determination, has led to a marked improvement in the quality of image generation. A similar enhancement is also evident in videos which is shown in Table 2.

Models Composition	FID	IS	CLIP Score
Dalle-e3 (Regular)	9.30	133.46	26.69
Diffusion (Regular)	9.15	158.23	26.58
Midjourney (Regular)	11.23	163.20	25.91
GPT3.5-Script+Dalle-e3	9.78	153.43	29.58
GPT3.5-Script+Diffusion	8.63	168.90	30.57
GPT3.5-Script+Midjourney	10.81	174.45	29.32
GPT4-Script+Dalle-e3	8.53	159.12	29.84
GPT4-Script+Diffusion	8.32	169.97	30.73
GPT4-Script+Midjourney	10.26	178.14	29.75
`DreamFactory(GPT4)+Dalle-e3`	**6.57**	**160.94**	**30.76**
`DreamFactory(GPT4)+Diffusion`	**7.03**	169.71	**30.92**
`DreamFactory(GPT4)+Midjourney`	**7.15**	**178**	**30.39**

Table 1: The statistical analysis of Text2Image task. All models can generate higher-quality images after prompts augmentation, but the quality of the images generated by our framework stands out.

Models Composition	FVD	KVD
Runway (Regular)	1879	125
Stable Video (Regular)	3560	182
`DreamFactory+Runway`	**732**	**62**
`DreamFactory+Stable Video`	**1376**	**113**

Table 2: The statistical analysis of Image2Video task. The improvement of our framework for generating multi-scene long videos is remarkable.

4.2 Multi-scene Videos Evaluation Scores

Cross-Scene Face Distance Score - In the generation of sequential videos, addressing character consistency is paramount. Discrepancies in the appearance of characters can lead not only to poor visual perception but also to the audience's inability to understand the plot and content. Maintaining character consistency ensures the coherence of the storyline revolving around the characters and enhances the visual appeal of the video. Especially, in the domain of long-duration videos, a video is typically composed of multiple scenes. This represents an unprecedented area of research, where there is a pressing need for robust evaluation metrics to assess the consistency of characters appearing across complex, multi-scene videos. Against this backdrop, we experimentally introduce the concept of the Cross-Scene Face Distance Score(**CSFD Score**), aimed at validating the issue of character facial feature consistency across different scenes.

In the computational process, each keyframe corresponds to a face, and using the dlib library, the position of the face can be extracted. The face-recognition library can be used to calculate the similarity score. For the facial segment of each frame, we can compute its similarity with all subsequent frames and then take the average. By this method, we can accurately determine whether the faces in the video are consistent. The relevant schematic diagram and the pseudocode for the calculation are provided in Algorithm 1.

Cross-Scene Style Consistency Score - In the production of long videos, maintaining stylistic consistency is equally important. A consistent style makes the video appear as a cohesive whole. Based on this concept, we have introduced the Cross-Scene Style Consistency Score(**CSSC Score**). However, to my knowledge, there currently isn't a mature method to rapidly determine the style of a video, so at this stage, we will rely on the assistance of large language-visual models. Essentially, we divide the video into several categories, which include: **anime, illustration, origami, oil painting, realism, cyberpunk, and ink wash**.

The calculation method for the Cross-Scene Style Consistency Score is as follows: For each key frame, a divider played by a GPT-4V is used to determine the classification. Once all scenes have been clearly divided into categories, the proportion of the most numerous category to the total number of key frames is calculated. Figure 6 presents a partial output where the input is "an elderly person making a traditional Chinese lantern in real life". Scene 4 depicts an animated lantern created using Dalle, with GPT-4V serving as the discriminator. It is observable that among the four scenes, the first three are categorized under a realistic style, while the fourth scene is classified as anime style. Consequently, the maximum number of distinct styles is three, resulting in a cross-scene style consistency score of 75%. The other relevant schematic diagram and the pseudocode for the calculation are provided in Algorithm 2.

Average Key-Frames CLIP Score - In the generation of long videos with multiple scenes, it is crucial to assess the alignment of each scene's keyframes with the corresponding text. They have incorporated a significant amount of additional information to ensure consistency, which could likely lead to deviations from the text during generation. This may result in the overall video not adhering to the script. Therefore, in this section, we propose the Average Key-Frames CLIP Score to ensure the consistency of key frame scenes with the script.

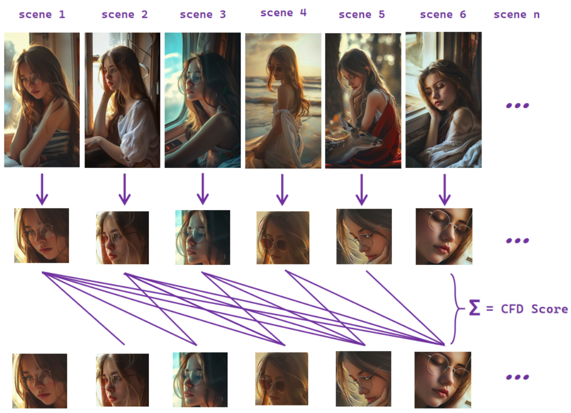

Figure 5: Schematic diagram and pseudocode for the calculation of Cross-Scene Face Distance Score.

Algorithm 1 Calculate CSFD Score

$total \leftarrow 0$
$count \leftarrow$ n*(n-1) / 2
for $i \leftarrow 1$ to n **do**
  for $j \leftarrow i+1$ to n **do**
    $similarity \leftarrow \text{CFS}(F_i, F_j)$
    $total \leftarrow total + similarity$
  end for
end for
$averageScore \leftarrow total/count$
return $averageScore$

Models	CSFD Score	CSSC Score	av-Clip Score
GPT4-Script+Dalle-e3	0.77	0.85	0.29
GPT4-Script+Diffusion	0.75	0.83	0.28
GPT4-Script+Midjourney	0.68	0.66	0.26
DreamFactory(GPT4)+Dalle-e3	**0.89**	**0.97**	**0.31**

Table 3: The statistical analysis of cross-scene score on different models.

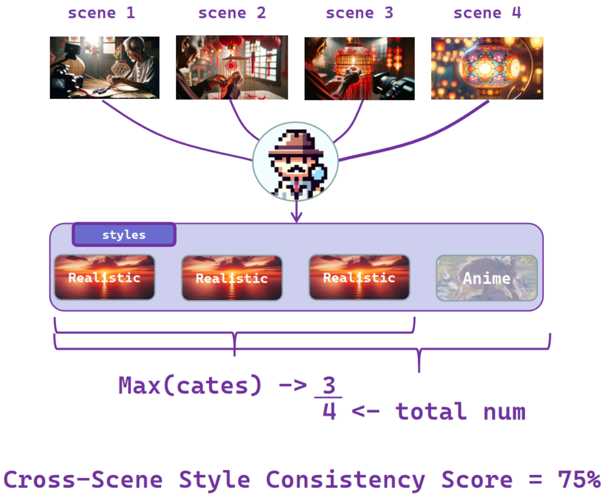

Figure 6: Schematic diagram and pseudocode for the calculation of Cross-Scene Style Consistency Score.

Algorithm 2 Calculate CSSC Score

$n \leftarrow$ number of key frames
$categories \leftarrow$ array initialized to 0 of size number of categories
for $i \leftarrow 1$ to n **do**
  $category \leftarrow \text{JUDGE}(F_i)$
  $categories[category] \leftarrow categories[category] + 1$
end for
$maxCount \leftarrow \max(categories)$
$crossSceneStyleScore \leftarrow \frac{maxCount}{n} \times 100$
return $crossSceneStyleScore$

The calculation method is straightforward: compute the CLIP score for each keyframe against the scene generated during scene prompting and take the average.

Results - In table 3, our data selection comprised seventy character-centric entries from the Multi-Scene Videos Dataset, produced by `DreamFactory` + GPT-4 + DALL-E 3. The baseline utilizes the DALL-E 3 model with script inputs from this segment. Furthermore, evaluations were conducted on the aforementioned (1) cross-scene facial distance, (2) cross-scene style scores, and (3) average CLIP Score. These metrics were used to assess the consistency of facial features within our framework, the consistency of scene attributes, and the alignment between prompts generated by our framework and the narrative, as well as imagery.

In our Cross-Scene facial distance scoring experiment, we employed the face locations method from the face-recognition library to locate 68 facial landmarks, thereby focusing the portrait photographs on the facial area. During the image encoding phase, we utilized the ViT model from the openai-clip repository to input the facial region and compute the vector representations. Subsequently, a vector dot product operation was performed to determine the final facial distance score. Owing to the inherent similarity among the facial images, all the scores were predominantly above 0.5. The specific reference facial match-score pairs are exhibited in Figure 7. In the analysis of both the CSSC score and the average CLIP score, the same set of seventy random samples was utilized as data. The CSSC Score employed GPT-4 Version as the stylistic analyst.

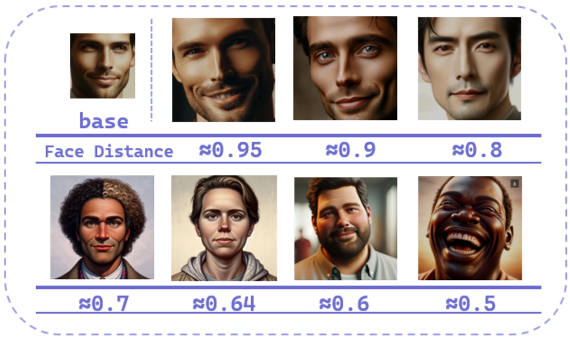

Figure 7: The distance between different faces when using openai-clip as the encoder.

5 Conclusion

We introduce **Dream Factory**: a multi-agent-based framework for generating long videos with multiple scenes. Dream Factory incorporates the idea of multi-agents into the field of video generation, producing consistent, continuous, and engaging long videos within the constraints of current computing power and model capabilities. Dream Factory introduces a keyframe iteration method to ensure alignment of style, characters, and scenes across different frames and can be built on top of any image or video generation tool. Furthermore, Dream Factory proposes new metrics to validate its capabilities by measuring the quality of generated content through cross-scene face and style consistency, as well as text-to-visual alignment. The evaluation of Dream Factory's work includes scores from over 20,000 real-life evaluations, culminating in the **Multi-Scene Videos Dataset**, which will be fully open source after acceptance.

6 Limitations

In this paper, we present a multi-agent video generation framework capable of producing videos with high consistency across multiple scenes and plotlines. However, we still face several limitations. Firstly, our current reliance on prompts to control agents means that the agents are not capable of highly creative tasks, such as devising plots with artistic merit. Such tasks require the accumulation of specific datasets for model fine-tuning. Secondly, the editing of all video segments is centered around synthesized speech content, which results in a final product that may appear as a mere assembly of clips. This necessitates the introduction of a unique framework design to enhance the fluidity of the videos. Lastly, video generation still involves substantial resource consumption.

7 Ethics Statements

The development and deployment of DreamFactory, a multi-agent framework for long video generation, raise several ethical considerations that must be addressed. The potential for the misuse of generated videos, such as the creation of deepfakes or the propagation of misinformation, is a significant concern. To mitigate these risks, we commit to implementing robust safeguards, including watermarking generated content and collaborating with fact-checking organizations. Additionally, we will ensure transparency in our research and make our methods and datasets publicly available, subject to ethical use guidelines. We also recognize the importance of diversity and inclusion in the training data to prevent biases in the generated content. Finally, we will engage with the broader community to establish ethical standards for the use of AI-generated video content, promoting responsible innovation and use of this technology.

References

Roy Armes. 2006. On video. Routledge.

Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, Yuanzhen Li, Michael Rubinstein, Tomer Michaeli, Oliver Wang, Deqing Sun, Tali Dekel, and Inbar Mosseri. 2024. Lumiere: A space-time diffusion model for video generation. Preprint, arXiv:2401.12945.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. 2023a. Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. 2023b. Align your latents: High-resolution video synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 22563–22575.

Tim Brooks, Janne Hellsten, Miika Aittala, Ting-Chun Wang, Timo Aila, Jaakko Lehtinen, Ming-Yu Liu, Alexei Efros, and Tero Karras. 2022. Generating long videos of dynamic scenes. Advances in Neural Information Processing Systems, 35:31769–31781.

Claude. Claude3.

Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. 2023. Structure and content-guided video synthesis with diffusion models. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 7346–7356.

Yingqiang Ge, Wenyue Hua, Kai Mei, Juntao Tan, Shuyuan Xu, Zelong Li, Yongfeng Zhang, et al. 2024. Openagi: When llm meets domain experts. Advances in Neural Information Processing Systems, 36.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. Advances in neural information processing systems, 27.

Jiaxi Gu, Shicong Wang, Haoyu Zhao, Tianyi Lu, Xing Zhang, Zuxuan Wu, Songcen Xu, Wei Zhang, Yu-Gang Jiang, and Hang Xu. 2023. Reuse and diffuse: Iterative denoising for text-to-video generation. arXiv preprint arXiv:2309.03549.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. 2023. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. arXiv preprint arXiv:2307.04725.

Sonam Gupta, Arti Keshari, and Sukhendu Das. 2022. Rv-gan: Recurrent gan for unconditional video generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2024–2033.

Anne M Harris. 2016. Video as method. Oxford University Press.

Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. 2022. Latent video diffusion models for high-fidelity long video generation. arXiv preprint arXiv:2211.13221.

Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, et al. 2023. Metagpt: Meta programming for multi-agent collaborative framework. arXiv preprint arXiv:2308.00352.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2024. Camel: Communicative agents for" mind" exploration of large language model society. Advances in Neural Information Processing Systems, 36.

Martin Merkt, Sonja Weigand, Anke Heier, and Stephan Schwan. 2011. Learning with videos vs. learning with print: The role of interactive features. Learning and Instruction, 21(6):687–704.

meta. Llama-2.

Ali Modarressi, Ayyoob Imani, Mohsen Fayyaz, and Hinrich Schütze. 2023. Ret-llm: Towards a general read-write memory for large language models. arXiv preprint arXiv:2305.14322.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. arXiv preprint arXiv:2112.09332.

openai. a. Sora.

openai. b. Sora.

Bruce M Owen and Steven S Wildman. 1992. Video economics. La Editorial, UPR.

pika. Pika.

Chen Qian, Xin Cong, Cheng Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong Sun. 2023. Communicative agents for software development. arXiv preprint arXiv:2307.07924.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. arXiv preprint arXiv:2307.16789.

Zeeshan Rasheed, Muhammad Waseem, Mika Saari, Kari Systä, and Pekka Abrahamsson. 2024. Codepori: Large scale model for autonomous software development by using multi-agents. arXiv preprint arXiv:2402.01411.

runway. Runway.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2024. Toolformer: Language models can teach themselves to use tools. Advances in Neural Information Processing Systems, 36.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2024. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. Advances in Neural Information Processing Systems, 36.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. Advances in Neural Information Processing Systems, 36.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2020. Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456.

Daniel Tang, Zhenghan Chen, Kisub Kim, Yewei Song, Haoye Tian, Saad Ezzini, Yongfeng Huang, and Jacques Klein Tegawende F Bissyande. 2024. Collaborative agents for software engineering. arXiv preprint arXiv:2402.02172.

John C Tang and Ellen Isaacs. 1992. Why do users like video? studies of multimedia-supported collaboration. Computer Supported Cooperative Work (CSCW), 1:163–196.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023a. Voyager: An open-ended embodied agent with large language models. arXiv preprint arXiv:2305.16291.

Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. 2023b. Modelscope text-to-video technical report. arXiv preprint arXiv:2308.06571.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. 2024. Executable code actions elicit better llm agents. arXiv preprint arXiv:2402.01030.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems, 35:24824–24837.

Hui Yang, Sifu Yue, and Yunzhong He. 2023. Auto-gpt for online decision making: Benchmarks and additional opinions. arXiv preprint arXiv:2306.02224.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. Advances in Neural Information Processing Systems, 36.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. arXiv preprint arXiv:2210.03629.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Yonatan Bisk, Daniel Fried, Uri Alon, et al. 2023. Webarena: A realistic web environment for building autonomous agents. arXiv preprint arXiv:2307.13854.

A Appendix

A.1 DreamFactory Responsibility allocation

As shown in Figure 8, our `DreamFactory` framework utilizes multiple large language models (LLMs) to form a simulated animation company, taking on roles such as CEO, Director, and Creator. Given a story, they collaborate and create a video through social interaction and cooperation. This framework allows LLMs to simulate the real world by using small video generation models as tools to accomplish a massive task. As illustrated in Figure 8, under their collaboration, it is possible to generate a series of consistent, stable, multi-scene long videos as the plot progresses.

A.2 User Study

Quantitative evaluation of human preference for video is a complex and difficult proposition, so we employed human evaluators to verify the quality of multi-scene videos generated by our framework. We collected 150 multi-scene short videos generated by AI from the internet and compare them with videos from our framework. Through this approach, we aimed to assess whether our videos could achieve an advantage in human preferences compared to existing AI videos on the network.

In our study, We adopt the Two-alternative Forced Choice (2AFC) protocol, as used in previous works [(Blattmann et al., 2023a), (Blattmann et al., 2023b), (Bar-Tal et al., 2024)]. In this protocol, each participant will be randomly shown a pair of videos with the same story, one is a short video collected on web platforms and the other is generated by our framework. Participants were then asked to select the superior side on five metrics: role consistency, scene consistency, plot quality, storyboard fluency, and overall quality. We collected 1320 human scores for this study, utilizing schools, communities, and network platforms. As illustrated in Figure 9, our method was preferred better.

A.3 Case Study

Comprehensive Keyframe Count Statistics - The version currently provided to users is balanced between cost and user experience, using the Short generation mode, typically around ten scenes. The specific number is related to the user's task input. The length of videos generated using random prompts is shown in the figure 10.

Figure 8: This figure presents the responsibility allocation chart for all employees within the DreamFactory architecture. For each employee, the upper left corner displays their role and portrait, while the upper right corner outlines the stages of participation and their roles. The essential parts of the prompt are depicted below.

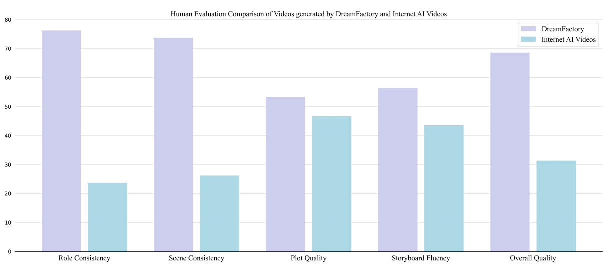

Figure 9: Human evaluation comparison of videos generated by DreamFactory and internet AI videos.

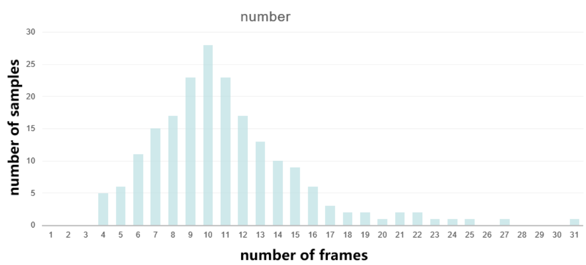

Figure 10: The key frame numbers count Statistics of DreamFactory.