
AesopAgent: Agent-driven Evolutionary System on Story-to-Video Production

Jiuniu Wang* Zehua Du* Yuyuan Zhao* Bo Yuan Kexiang Wang Jian Liang
Xaxi Zhao Yihen Lu Gengliang Li Junlong Gao Xin Tu Zhenyu Guo†

DAMO Academy, Alibaba Group

Abstract

The Agent and AIGC (Artificial Intelligence Generated Content) technologies have recently made significant progress. We propose **AesopAgent**, an **Agent-driven Evolutionary System on Story-to-Video Production**. AesopAgent is a practical application of agent technology for multimodal content generation. The system integrates multiple generative capabilities within a unified framework, so that individual users can leverage these modules easily. This innovative system would convert user story proposals into scripts, images, and audio, and then integrate these multimodal contents into videos. Additionally, the animating units (e.g., Gen-2 and Sora) could make the videos more infectious. The AesopAgent system could *orchestrate task workflow* for video generation, ensuring that the generated video is both rich in content and coherent. This system mainly contains two layers, i.e., the Horizontal Layer and the Utility Layer. In the Horizontal Layer, we introduce a novel *RAG-based evolutionary system* that optimizes the whole video generation workflow and the steps within the workflow. It continuously evolves and iteratively optimizes workflow by accumulating expert experience and professional knowledge, including optimizing the LLM prompts and utilities usage. The Utility Layer provides multiple utilities, leading to *consistent image generation* that is visually coherent in terms of composition, characters, and style. Meanwhile, it provides audio and special effects, integrating them into expressive and logically arranged videos. Overall, our AesopAgent achieves state-of-the-art performance compared with many previous works in visual storytelling. Our AesopAgent is designed for convenient service for individual users, which is available on the following page: <https://aesopai.github.io/>.

1 Introduction

The recent development of base Large Language Models (LLMs) [3, 38, 61, 63], and Multimodal Models [69, 33, 39, 30] has catalyzed significant changes in Artificial Intelligence Generated Content (AIGC). This advancement has led to the effective integration of generative AI technology with traditional Professional Generated Content (PGC) and User Generated Content (UGC), addressing a wide range of user requirements. Notably, technologies such as Stable Diffusion [46], DALL-E 3 [2], and ControlNet [70] have excelled in generating and editing high-quality images, attracting significant interests in both academy and industry. Video generation, in contrast to simple static image generation, necessitates managing complex semantic and temporal information, presenting great challenges. Recent initiatives, such as Make-A-Video [50], Imagen Video [17], PikaLab [27], and Gen-2 [?], have demonstrated the ability to produce short videos from textual descriptions. And Sora [40] is capable of generating high-definition videos up to one minute in length. However, story-to-video

*Equal Contribution.

†Corresponding author.



Figure 1: **Overview of our AesopAgent.** This system would convert the user story proposal into a video assembled with images, audio, narration, and special effects. The video generation workflow suggested by AesopAgent, utilizing agent-based approaches and RAG techniques, encompasses script generation, image generation, and video assembly.

production requires the cooperation of various modules, so gaps remain in image expressiveness, narrative quality, and user engagement. To bridge these gaps, we propose AesopAgent, an agent-driven evolutionary system on story-to-video production. With agent technology, we could effectively harness advanced visual and innovative narrative generation technologies to tackle the complex video generation task, thereby creating videos with compelling narratives and attractive visual effects. Our system could understand user intentions and identify suitable AI utilities to fulfill user requirements. This feature enables users to effortlessly employ AI capabilities.

AesopAgent utilizes agent-based approaches and RAG techniques, coupled with the incorporation of expert insights, to facilitate an iterative evolutionary process that results in an efficient workflow. This workflow creates high-quality videos from user story proposals automatically. As illustrated in Figure 1, upon receiving a user’s “Dragon Story” proposal, AesopAgent employs the well-designed workflow by agents implemented with RAGs, including script generation, image generation, and video assembly, ultimately generates a high-quality dragon story video.

In this paper, our system features a comprehensive process orchestration, including a Horizontal Layer and a Utility Layer. The Horizontal Layer systematically executes high-level strategic management and optimization in the video generation workflow continuously and iteratively. The Utility Layer provides a suite of fully functional utilities for each task of the workflow, ensuring the effective execution of the video generation process. In the Horizontal Layer, we introduce two types of Retrieval-Augmented Generation (RAG) techniques[11]: Knowledge-RAG and Experience-RAG (denoted as K-RAG and E-RAG). E-RAG collects expert experience, and K-RAG leverages existing professional knowledge. Our system constructs and updates the database of E-RAG and K-RAG with the guidance of experts, leading to our evolution ability. The Utility Layer assembles multiple utilities based on each unit’s fundamental capabilities, optimizing their usage based on feedback to ensure stable, rational image composition in terms of character portrayal and style.

The effectiveness of our system is evaluated in three key points: storytelling, image expressiveness, and user engagement. A comparative analysis with systems like ComicAI [15] and Artflow [14] shows that AesopAgent achieves state-of-the-art visual storytelling ability, excelling in coherence,

representation, and logic through RAG techniques. Manual evaluations against ComicAI [15] in image element restoration, rationality, and composition indicate AesopAgent’s superiority. When compared with parallel video generation research such as NUWA-XL [64] and AutoStory [59], our system demonstrates leading performance in image complexity and narrative depth. Furthermore, AesopAgent’s adaptability is evident in its ability to integrate external software like Runway, catering to specific user needs and underscoring its extensive scalability.

In summary, the contributions of our AesopAgent are threefold:

- 1) We have leveraged agent techniques to automate *task workflow orchestration* for video generation, enabling the effective process of converting story proposals into final videos. The workflow is designed by the agent and includes three main steps (script generation, image generation, and video assembly).
- 2) We proposed a *RAG-based evolutionary system*, collecting expert experience (E-RAG) and professional knowledge (K-RAG) to improve the system performance. The prompts of LLM would be optimized iteratively according to the experience and knowledge in the RAG database, generating narrative coherence text (e.g., script, image descriptions). The RAG techniques also guide the utilities usage, consequently enhancing the visual quality of the generated videos.
- 3) We achieve *consistent image generation* since our AesopAgent skillfully applies specialized utilities, thus the generated images maintain a cohesive and professional visual expression throughout the shots. The main components of *consistent image generation* are the image composition module, the multiple characters consistency module, and the image style consistency module. There are many utilities in these modules, and some of them have been optimized by our AesopAgent.

2 Related Works

2.1 AI Agent

Recent advancements in AI agent technology stand at the vanguard of transformative developments within the artificial intelligence domain. A plethora of commendable works involving AI agents have emerged, showcasing groundbreaking progress and innovation. Notable works include: OpenAGI [12], AutoGPT [62], Voyager [57], RET-LLM [37], ChatDev [42], AutoGen [60], etc. These agents exhibit diverse performances in aspects like memory function configuration, task planning capabilities, and modifications to the base Large Language Model (LLM).

As for memory function configuration, recent multi-agent systems (e.g., AutoGen [60], MetaGPT [19], ChatDev [42], AutoGPT [62]) adopted a shared memory pool for top-level information synchronization, with different technical implementation solutions. Specifically, MetaGPT [19] use sub/pub mechanisms, AutoGen [60] use dialogue delivery. The single-agent systems (e.g., Voyager [57], GITM [74], ChatDB [21]) have consensus capabilities (reading, writing, and thinking), combining long-term and short-term memory. In this context, long-term memory structures mostly use historical decision sequences, memory streams, or vector databases, while short-term memory is primarily based on contextual information.

With regard to task planning and utility usage capabilities. The technical solutions for planning capabilities range from prompts and chain mechanisms to standard operating procedures (SOP). Some models, such as Generative Agents and MetaGPT [19], enhance execution accuracy and controllability through refined planning and structured output, while models like CAMEL [29] and ViperGPT [51] focus on using historical information and context to guide the behavior of the Agents. AutoGPT [62] and AutoGen [60] models emphasize dialogue and cooperation among multiple Agents to accomplish more complex tasks.

When it comes to modifications of the base model (LLM), The majority of agents (e.g., CAMEL [29], ViperGPT [51], AutoGPT [62], ChatDev [42], MetaGPT [19], AutoGen [60]) did not modify the basic LLM, indicating a current preference for maintaining its original structure and enhancing performance through peripheral enhancements. While some agents (e.g., ToolLLM [43], OpenAGI [12]) implemented SFT in the LLM. Particularly for ancillary enhancements, Retrieval-Augmented Generation (RAG) [11, 34] technology has gained increasing traction in fortifying Large Language Models (LLMs). RAG amplifies response accuracy[67, 65] and pertinence by dynamically integrating information from external knowledge bases into LLM responses [71, 68], effectively mitigating issues such

as hallucinations [24]. Its applicability in devising more pragmatic chatbots and applications like QA, summarization, recommendation [10, 5, 44] has driven rapid technological growth, encompassing methodological and infrastructural domains.

2.2 Generative Models

Generative AI models refer to artificial intelligence models capable of autonomously generating new data, such as text, images, video, and audio. They have affected many tasks, including those in natural language processing (NLP), computer vision, and multimodal domains. In natural language processing, the GPT series, notably GPT-3 [3] and GPT-4 [38], have revolutionized our comprehension and generation of natural language. These advanced models have given rise to acclaimed AI systems, such as X-GPT [61], and MM-REACT [63]. In computer vision, generative models are primarily utilized for enhancing image and video generation. Initial approaches often employed techniques like VAE [26], GAN [13], VQVAE [53], or transformers [54]. Diffusion models (DDPM) [18] and vision transformer (ViT) [8] benefit for producing high-quality images and videos. The diffusion models, such as Stable Diffusion [46] and Stable Diffusion XL [41], can generate images by denoising certain steps from random noise. Vision transformer (ViT) [8] based models like CogView2 [7] and Parti [66] can generate image tokens autoregressively. Among these image synthesis models, DALL-E 3 [2] has superior performance, which can generate high-resolution images and excel in converting textual descriptions into vivid images. There are some recent works [50, 17, 58] on video synthesis, and they often employ temporal modules [50] (i.e., temporal convolution and temporal attention) to generate several frames simultaneously. In the multimodal domain, previous works like NExT-Chat [69], LLaVA [33], GPT-4V [39], and BLIP-2 [30] harmonize language and visual elements, supporting for creating complex and enriched content. Additionally, there are some models (e.g., Audiocraft [6], MusicLM [1], and Music Transformer [23]) for music generation, demonstrating the adaptability of transformers in handling intricate structures in audio sequences.

2.3 Story Visualization

The story visualization task [4, 28, 31, 36] aims to generate visually consistent images or videos from a story described in neural language. Previous story visualization methods mainly focus on conditional text-to-image synthesis [45, 25, 73], which can generate high-resolution realistic images. The core of this task is to get the high-quality keyframes conditioned by the story proposals. For example, the input is a complete dialogue session instead of a single sentence in [49], and the model is required to generate a visual story. Recently, video generation has been adapted to the story visualization, especially text-to-video [32] or image-to-video generation [52, 56]. The challenge in video generation for story visualization is to use visual impact to vividly and interestingly describe the given story. DirecT2V [20] and Phenaki [55] generate different video frames based on various sentences as conditions, forming short video clips. VideoDrafter [35] and Vlogger [75] could convert user proposals into scene descriptions, then generate a video clip for each scene. Given key frames of a video, NUWA-XL [64] uses the frame interpolating unit to create a video of arbitrary length. Additionally, some systems, like Artflow [14] and ComicAI [15], could convert user story proposals into videos containing images and audio. However, there is no complete system that can fully display the keyframes, audio, and special effects that are full of story tension in a video. These elements are important for the video to express the story.

3 Methodology

3.1 Overall Framework

The AesopAgent accepts a user story proposal as input and through the implementation of a video generation workflow, it creates a video that conveys the story. As depicted in Figure 2, the architecture of the AesopAgent comprises two layers: the Horizontal Layer and the Utility Layer.

The Horizontal Layer employs agent techniques to systematically execute high-level strategic management and optimization in the video generation workflow continuously and iteratively. The expert experience, as well as professional knowledge, are aggregated to construct E-RAG and K-RAG. Then E-RAG and K-RAG are utilized to enhance LLM prompts, so that the Horizontal Layer can employ

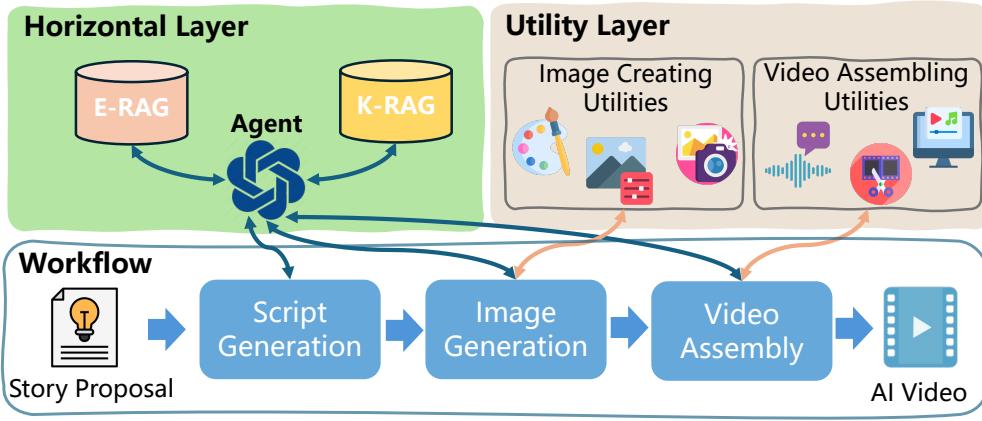


Figure 2: **Illustration of the AesopAgent framework.** The bottom part of the figure shows the workflow from the user story proposal to the video, and the top part shows the main components of our method: the Horizontal Layer and the Utility Layer. The Horizontal Layer is responsible for leveraging agent and RAG techniques, optimizing workflow and prompts, and optimizing utilities usage, and the Utility Layer is responsible for providing utilities for image generation and video assembling steps.

K-RAG and E-RAG to create and continuously optimize the workflow, further raising requirements for utilities in the Utility Layer and optimizing utilities usage.

The Utility Layer provides a suite of fully functional utilities for each step of the workflow, tailored to task-specific requirements, thus ensuring the effective execution of the video generation process. This layer primarily includes utilities for creating images and utilities for assembling videos. We will introduce the details of the Horizontal Layer and Utility Layer in the following subsections.

3.2 Horizontal Layer

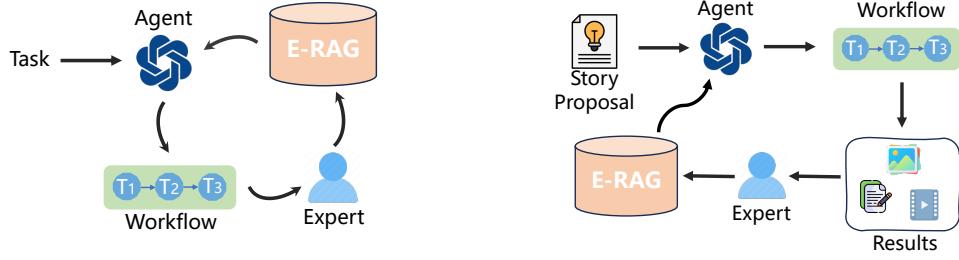
In the Horizontal Layer, we leverage agent technology to facilitate workflow creation and optimization with the tasks in the video generation workflow. Our agent \mathcal{M} has the abilities of workflow creating and task planning \mathcal{M}_T , prompt execution \mathcal{M}_e , and database updating \mathcal{M}_u . Our agent combines the accumulated expert experience E and professional knowledge K within the video generation process through interactions with experts. We have developed an efficient and feasible video generation workflow T , which is primarily divided into script generation step, image generation step, and video assembly step.

Moreover, our agents also optimize the prompts to enhance the quality of script generation and also improve utilites usage skills. We propose *RAG-based evolutionary* method (specifically K-RAG and E-RAG) to support these functional parts. K-RAG stores expert knowledge to guide the creation of more professional scripts and utilities usage, while E-RAG summarizes feedback from experts and users, integrating these experiences to refine the video generation workflow. To the best of our knowledge, our AesopAgent is the first system to support task planning and prompt optimization ability based on the RAG techniques. In the following sections, we will progressively expound on the specifics of each module.

3.2.1 Task Workflow Orchestration Module

The process from a story proposal to a fully realized video is a multifaceted work that encompasses numerous AI methodologies and considerable effort. Establishing an effective task execution workflow is essential for achieving optimal results and enhancing efficiency. Through interaction with agents and leveraging RAG techniques, we collect expert feedback to iteratively optimize the execution workflow, finally, a highly efficient workflow was established.

Our methodology adopts a two-stage strategy for workflow creation and optimization. In the first stage, given a task description D_T and task planning prompt x_T , and a task planing agent (LLM)



(a) First stage of workflow creation and optimization. (b) Second stage of workflow creation and optimization.

Figure 3: The two stages of workflow creation and optimization. In the first stage, we interact with the agent about a feasible workflow scheme with E-RAG. In the second stage, when we get a feasible workflow scheme, we continue to optimize workflow by combining the results of actual implementation with E-RAG.

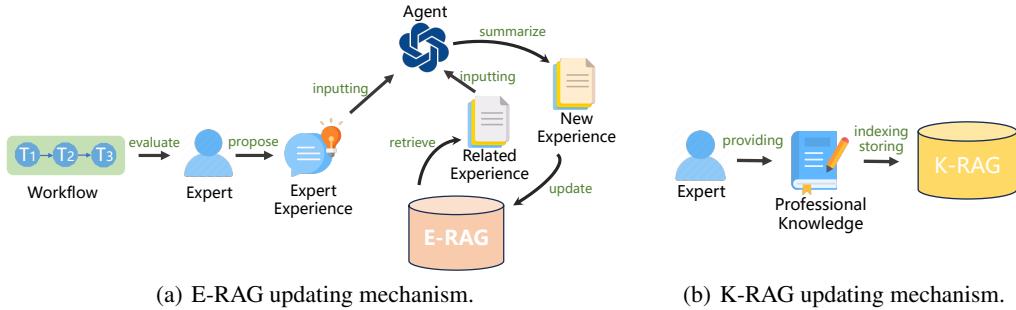


Figure 4: Two types of RAG updating mechanism. The left part shows the E-RAG updating process by taking the optimization workflow as an example. The agent summarizes expert experience and relevant experience in the E-RAG dataset, and writes the new experience to the E-RAG database to implement updates. The right part shows the updating process of K-RAG. The K-RAG database is updated by indexing and storing the knowledge documents provided by experts.

\mathcal{M}_T , we execute \mathcal{M}_T with D_T and x_T as inputs, i.e. $\mathcal{M}_T(D_T, x_T)$, to generate an initial workflow $T = (T_1, \dots, T_n)$. Subsequently, this workflow undergoes expert evaluation to gauge its practicality, and specific suggestions are provided by experts. E-RAG updates experience to refine the task decomposition process. As the experience of E-RAG accumulates, we can use E-RAG to optimize the workflow. Given experience from RAG: $E = \{e_1, \dots, e_n\}$ and Retriever \mathcal{R} , according to the task T , retrieve relevant experience e_i , the agent combines prompt x_T and experience e_i to obtain the execution result r from $\mathcal{M}_T(x_T, e_i)$. In the second stage, once a feasible task segmentation is established, we gather manual feedback from actual task execution results of tasks(such as scripts, images, videos, etc.). E-RAG then utilizes this feedback to further update its experience database and optimize the workflow. The two stages are illustrated in Figure 3.

The E-RAG update process is as follows: experts provide feedback based on the result, denoted as s . Relying on s , the system retrieves the most pertinent experience e_j . Subsequently, the agent generates novel experience by synthesizing the feedback and prior experience, represented as \hat{e}_j with agent $\mathcal{M}_u(s, e_j)$, then updating the database of experiences e_j with \hat{e}_j accordingly. If there is no similar experience in the RAG database, the agent formulates a new experience based on the feedback and records it as e_{n+1} using $\mathcal{M}_u(s)$. The E-RAG updating mechanism is illustrated in Figure 4(a).

Through the iteration of the above processes, we finally get the following workflow, which mainly contains three steps, script generation step, image generation step, and video assembly step. The detailed workflow is in Figure 5. Note that we do not have the optimal workflow, in fact, continuing to run our system and gain experience will make our workflow more reasonable, and our AesopAgent system is constantly improving.

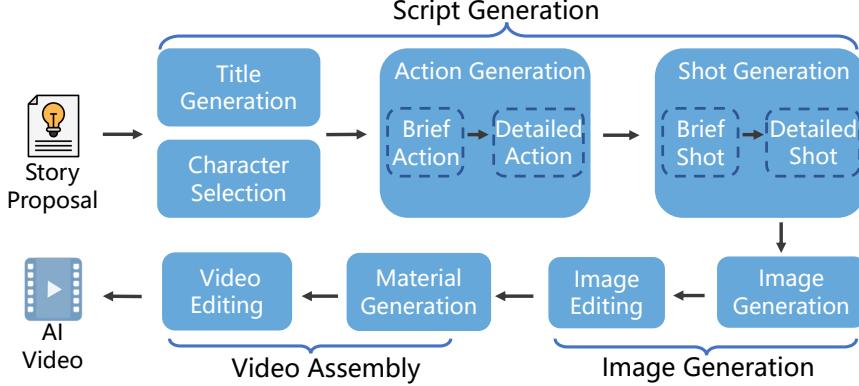


Figure 5: The overall workflow of story-to-video production. The workflow mainly contains three steps: script generation, image generation, and video assembly. The script generation step includes the title generation process and character design and selection process, followed by the action generation and shot generation process. The image generation step mainly includes the image generation process and image editing process. The video assembly step mainly includes the video material generation process and video editing process.

3.2.2 Prompt Optimization Module

Our goal is to produce high-quality, captivating storytelling videos. Thus, a professionally crafted script is crucial to achieving the outcomes. In this step, the quality of prompts plays a crucial role. We employ agent technology to iteratively enhance the prompts of the script generation step, culminating in well-structured, narrative-rich scripts that underpin an effective video generation workflow. In this module, we utilize K-RAG to integrate expert knowledge into the prompts of the scriptwriting process and E-RAG to refine prompts based on the feedback from actual results.

Providing Professional Knowledge. The creation of videos with strong storytelling appeal involves numerous professional intricacies, such as shot composition, scene framing, and dialogue arrangement. We deploy K-RAG techniques to offer insights from professional screenwriting and filming practices. The utilize of K-RAG is: given an script prompt x_s and a set of knowledge documents from $\text{RAG}:K = \{k_1, \dots, k_n\}$, the Retriever \mathcal{R} , retrieves relevant knowledge k_i . Subsequently, the agent generates result r from $\mathcal{M}_e(x_s, k_i)$ based on script prompt \hat{x}_s which is x_s concatenated with domain expertise k_i . The updating mechanism of K-RAG is depicted in Figure 4(b), experts provide relevant knowledge documents, and the RAG system performs indexing and storing to write the knowledge into the K-RAG database.

Prompt Optimization Based on Feedback. Large Language Models (LLMs) have significant potential for task execution; however, their performance is heavily reliant on the quality of prompts. In general, the most efficacious prompts are manually crafted through iterative human refinement. In this paper, we propose to use E-RAG to optimize prompts, which can assimilate human experience iteratively. Given a prompt x and a set of experience documents $E = \{e_1, e_2, \dots, e_n\}$ the Retriever \mathcal{R} retrieves pertinent experience e_i , which facilitates the generation of an optimized script prompt: \hat{x}_s from $\mathcal{M}_e(x_s, e_i)$. The updating mechanism of E-RAG parallels the mechanism depicted in Figure 4(a).

With the K-RAG and E-RAG, we iteratively refine the prompts of the script generation effect and ultimately succeed in generating scripts that are both high-quality and engaging for storytelling. As shown in the script generation step in Figure 5, we could generate the title, select the appropriate characters, and generate information on actions and shots. Here action refers to a shooting scene, generally an action contains several shots.

3.2.3 Utilities Usage Module

Given the generated script, the agent system must invoke a suite of utilities to execute the video production, including image generation and video assembly process. Effective usage of these utilities is crucial for achieving high-quality video output.

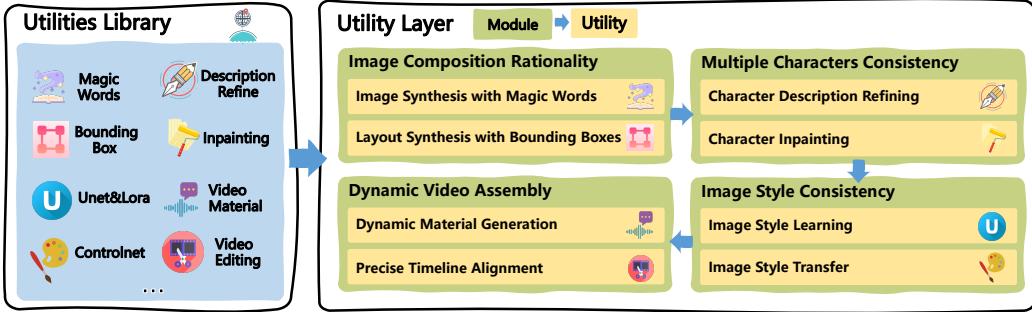


Figure 6: **The illustration of Utility Layer.** The Utility Layer contains four modules, i.e., image composition rationality, multiple characters consistency, image style consistency, and dynamic video assembly. Each module has some utilities from the utilities library, and these utilities are created and optimized by the Horizontal Layer.

Utility Suggestion & Creation. We explore the utilities usage to guide the Utility Layer with K-RAG. We expect that the agent system can understand the provided utilities and actively seek utilities according to requirements. To achieve this, we first provide some basic utilities and use tutorials according to the workflow, and we write the knowledge into the K-RAG database. When the agent performs a specific task, it retrieves the necessary utilities from K-RAG as required. The agent evaluates whether the current utilities can fulfill the task by comparing the task goals against the utilities usage instructions. If the available utilities are insufficient, the agent could find some related utilities from the web according to the task requirements.

Optimization of Utilities Usage. We optimize the utilities usage with E-RAG. Similar to the prompt optimization process, we expect agents to continually optimize their utility usage based on feedback. Taking image utility usage as an example, we use the E-RAG to optimize the image generation workflow. Given image descriptions d^I and experience $\{e_1, e_2, \dots, e_n\}$ from E-RAG database, retriever \mathcal{R} , according to the image descriptions d^I , the retriever selects the most relevant experience e_i . The image generation process generates images $\{i_1^g, \dots, i_N^g\}$ using $\mathcal{G}(d^I, e_i)$, where N is the number of shots for the generated video and $\mathcal{G}(\cdot)$ is the image generation function (containing DALL-E 3 [2] or SDXL [41]). We evaluate the image against the requirements using manual assessment and multi-modal model(e.g., GPT-4V [39]), modification suggestions s are obtained, which guide the retrieval of experience e_j . The agent synthesizes new experience \hat{e}_j from $\mathcal{M}_u(s, e_j)$ using the given suggestions and prior experiences, and subsequently updates the experience in the E-RAG database. Through iteration, we progressively accumulate better image generation experience, thus acquiring more images that align with our objectives.

3.3 Utility Layer

Given the user’s story proposal, our system designs a script of N shots with image descriptions $d^I = \{d_1^I, \dots, d_N^I\}$. We aim to generate images $\{i_1^g, \dots, i_N^g\}$ that match the image descriptions, then assemble video material (i.e., images, audio, and narrates) into a dynamic video to show a complete story. As shown in Figure 6, the Utility layer contains four modules, i.e., image composition module, multiple characters module, image style module, and video assemble module. We propose an innovative approach to integrate utilities into these four modules. The former three modules contribute to our *consistent image generation*.

3.3.1 Image Composition Rationality

The image composition module aims to generate images $\{i_1^{co}, \dots, i_N^{co}\}$ with composition rationality. This module is implemented with the following two utilities, i.e., image synthesis with magic words and layout synthesis with bounding boxes.

Image Synthesis with Magic Words. We employ an image synthesis unit to convert the image description d_t^I to a well-composed image i_t^{co} for the t -th shot of the video. During the training process of the image synthesis unit, certain “magic words” have been annotated within the training dataset, enabling the image synthesis unit to comprehend their meanings. Then these composition-related

magic words are used as prefixes to generate images that meet specific compositional requirements. Typical magic words include “Middle view”, “Close view”, “Low Angle”, “High Angle”, etc.

Layout Synthesis with Bounding Boxes. To fulfill more detailed compositional demands, the t -th image i_t^{co} could also employ bounding boxes $\{box_1^t, \dots, box_{k_b}^t\}$ as another controllable condition, where k_b is the number of bounding boxes for the t -th image. This allows for generating objects in fixed locations corresponding to the given descriptions. This approach, known as Layout Diffusion [72], can put different characters into specified locations to create a cohesive scene within one image.

3.3.2 Multiple Characters Consistency

The multiple characters consistency module aims to ensure the consistent characters appearance for the images of the same story. We provide two utilities in this module, i.e., character description refining and character inpainting. So that the multiple characters module turns the well-composed images $\{i_1^{co}, \dots, i_N^{co}\}$ into the characters consistent images $\{i_1^{ch}, \dots, i_N^{ch}\}$.

Character Description Refining. Our approach refines two types of character descriptions for consistency: attached and separate character descriptions. The attached character description, embedded within the overall image description, encompasses key characteristics such as gender, age, clothing, and hairstyle. This integration ensures the generation of similar characters in different images based on the attached character description within the t -th shot’s image description d_t^I . Additionally, to portray characters more specifically, we generate separate character descriptions $d_t^{ch} = \{d_{t,1}^{ch}, \dots, d_{t,k_c}^{ch}\}$ for each main character in the story as the semantic condition for Character Inpainting, where k_c is the number of main characters in the t -th shot.

Character Inpainting. We segment the i_t^{co} , and obtain the region mask $m_t^{ch} = \{m_{t,1}^{ch}, \dots, m_{t,k_c}^{ch}\}$ for k_c characters. Subsequently, the character inpainting is performed on the image at each position of m_t^{ch} to obtain an image i_t^{ch} that better aligns with the separate character description d_t^{ch} . This process can be expressed as

$$i_t^{ch} = \text{Inp}(i_t^{co}, m_t^{ch}, d_t^{ch}), \quad (1)$$

where $\text{Inp}(\cdot)$ represents the character inpainting unit, re-illustrating the character at m_t^{ch} based on the separate character description d_t^{ch} . The diffusion model is the core of the character inpainting unit, and we employ technology such as DreamBooth [48] and the parameter fusion of multiple models to gain consistent character generation.

3.3.3 Image Style Consistency

The multiple characters module generate character consistent images $\{i_1^{ch}, \dots, i_N^{ch}\}$. The image style module makes these images uniform in style, gaining the final generated images $\{i_1^g, \dots, i_N^g\}$. We choose two utilities to support image style consistency, i.e., image style learning and image style transfer.

Image Style Learning. To enable the image synthesis unit to generate images in a specific style s_{ty} (e.g., colorful comic style), we first collect a set of images in that style for training, allowing the unit to incorporate the style into its parameters of deep neural networks. In detail, these parameters could be the UNet [47] or LoRA [22] layers. The image synthesis unit is trained with the collected image-text pairs. Once trained with images of a fixed style, the unit can consistently generate images in a certain style.

Image Style Transfer. Here the ControlNet [70] unit is employed for image-style transfer, so that the generated images could have the same style. We also obtain the depth map m_t^{dp} of image i_t^{ch} as a condition. So the ControlNet [70] unit $\text{CT}(\cdot)$ could be employed to transform the image i_t^{ch} and image description d_t^I into i_t^g as,

$$i_t^g = \text{CT}(s_{ty}, d_t^I, i_t^{ch}, m_t^{dp}, \lambda_{ct}), \quad (2)$$

where the hyperparameter λ_{ct} aims to balance the editing intensity, controlling the weights of different channels within ControlNet [70].

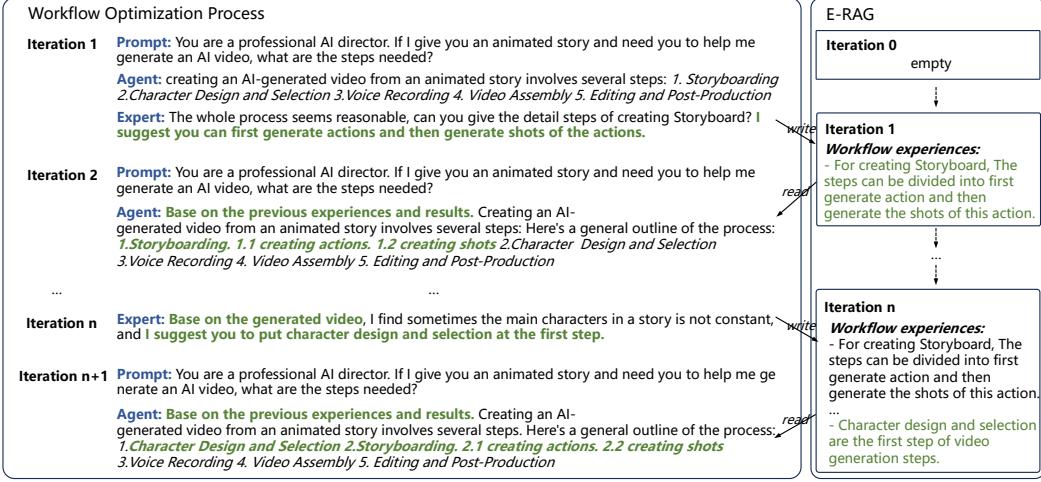


Figure 7: **Case of workflow optimization using E-RAG.** The text in green is the result of each step’s update, we note that by using E-RAG, the agent system has learned from experts’ experience for optimizing workflow.

3.3.4 Dynamic Video Assembly

The dynamic video assembly module could be split into two utilities, i.e., dynamic material generation and precise timeline alignment. Unlike traditional simple editing rules and template-based approaches, this module integrates video content with audio and transition effects.

Dynamic Material Generation. Dynamic material generation automatically selects and edits materials based on the narrative requirements, focusing on audio and video material generation. Our system incorporates background music, sound effects, and voice elements for audio, with the voice component generated using Text-to-Speech (TTS) technology. The goal of video material editing is to transform storyboard images into dynamic videos. This includes detailed descriptions and visual effect analyses (e.g., push and pull shots, rotations, zooms) and various transition effects (e.g., dissolves, wipes, and pushes).

Precise Timeline Alignment. Timeline alignment assembles different materials into a cohesive video. This utility ensures that all selected and edited materials are synchronized on the main timeline. The system employs precise timing adjustments to maintain video coherence, addressing the original material temporal misalignment during camera movement and transitions. These features simplify the editing process, enhancing system efficiency and video coherence.

4 Experiments

4.1 Main Results of Horizontal Layer

4.1.1 Task Workflow Orchestration

In this section, we discuss the enhancements that agent technology contributes to task workflow orchestration and optimization. We show this process through a case study.

Workflow Optimization. We use the E-RAG technique to iteratively optimize the process of workflow, as depicted in Figure 7. We illustrate our technology by presenting typical cases in the optimization process. At the beginning of the iteration, the E-RAG database is empty, and experts give suggestions for decomposing the storyboard into actions and shots according to the previous result. These suggestions are summarized by the agent and written into our E-RAG database. In the subsequent iteration, the agent leverages its accrued experience and previous results to propose a more reasonable workflow. After n iterations, we observed that the main characters in different actions of the produced videos lacked stability, so we proposed to put character design and selection

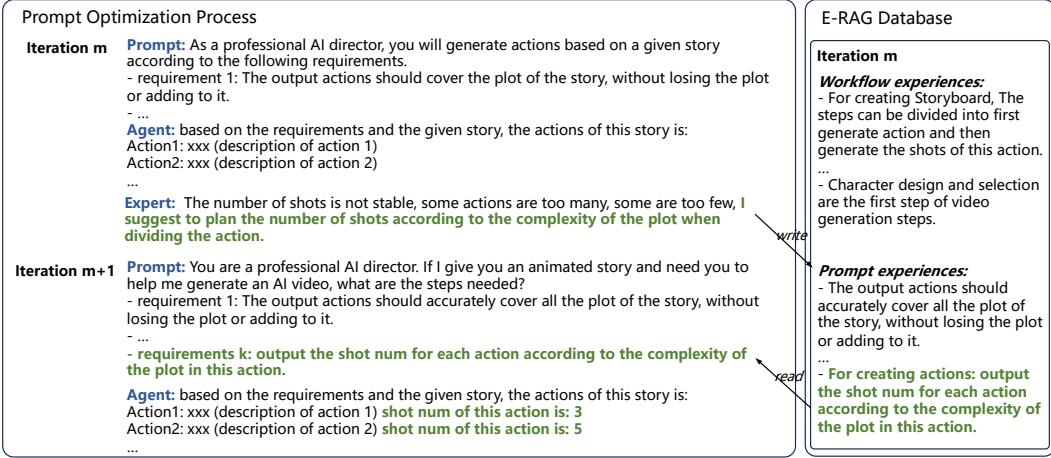


Figure 8: **Case of prompt optimization using E-RAG.** The text in green is the result of each step’s update, we note that by using E-RAG, the agent system has learned from experts’ experience for optimizing prompts.

| Methods | Completeness (30%) | Fidelity (30%) | Logical Coherence (40%) | Overall |
|---------------------------------------|--------------------|----------------|-------------------------|-------------|
| ComicAI [15] | 85 | 33 | 78 | 66.6 |
| Artflow [14] | 98 | 43 | 87 | 77.1 |
| AesopAgent w/o professional knowledge | 92 | 32 | 85 | 71.2 |
| AesopAgent (Ours) | 94 | 60 | 89 | 81.8 |

Table 1: **Script evaluation results.** The best results are shown in bold.

as the first step of the workflow to ensure the stability of the main characters throughout the story. Similarly, by utilizing E-RAG techniques, we successfully optimized the workflow.

4.1.2 Prompt Optimization and Script Evaluation

In this section, we show how RAG improves prompt optimization. We show the optimization process of prompt in the script generation step through a case study. We invite professional screenwriters to evaluate the scripts generated by our method to show the comparison between our professional knowledge of prompt optimization. We also show the comparison with other systems from the evaluation scores.

Prompt Optimization. In a similar vein, agent technology can be applied to the enhancement of prompt optimization, as illustrated in Figure 8. The expert observed that the allocation of shot numbers for each action did not correspond to the complexity of the story, thereby it is suggested to add global shot number planning during the action division step, which will make the generated shots more reasonable. Similarly, in the application of E-RAG techniques, we successfully implemented such enhancements in prompt optimization.

Script Evaluation. We engaged professional scriptwriters to evaluate our scripts. They established evaluation criteria for **completeness** (evaluate the completeness of the story structure, the completeness of the content in each shot etc.), **fidelity** (evaluate the degree of reduction of the story etc.), and **logical coherence** (evaluate whether the story plot and the arrangement of the script is logical coherence etc.) to conduct a comprehensive assessment of the script quality. We calculated an overall score based on the weighted sum of these three scores to gauge the overall quality of the script. We compared the results of our method with those obtained without professional knowledge (K-RAG), as well as with the best story creation applications currently available on the market: Artflow [14] and ComicAI [15]. Below are detailed analyses:

Comparison of Professional Knowledge. We first compared the scripts generated by our method with our method without expert knowledge (where K-RAG was not used during the script generation step). The absence of expert knowledge during the script generation process means there is a lack of guidance for crafting the scripts, which can lead to some unprofessional aspects such as scene

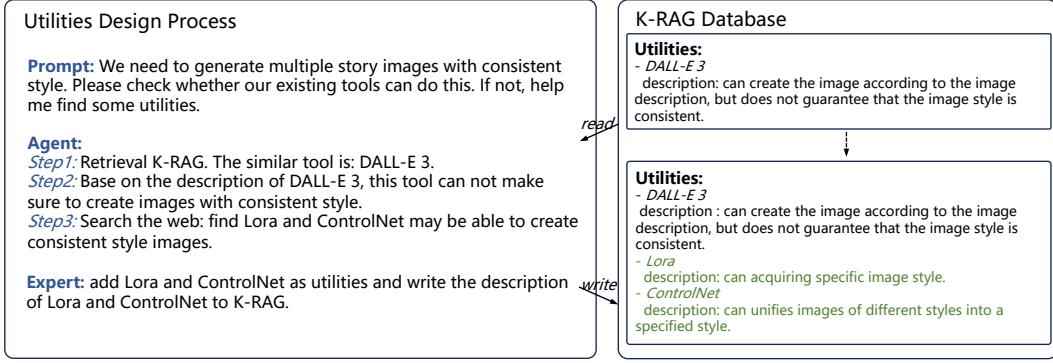


Figure 9: **Case of Utility Suggestion & Creation with K-RAG.** The text in green is the result of each step’s update, we combined K-RAG and agent technology to help us complete the utility design of style-consistent images.

arrangement and plot segmentation, resulting in poorer outcomes, as shown by the results in Table 1, there was a decline in completeness, fidelity, and logical coherence, particularly in fidelity. This decrease can be attributed to the lack of expert guidance, which makes the language model generate freely, leading to story error propagation and significantly reduced fidelity. This demonstrates the importance of professional knowledge (K-RAG) for the quality of script generation.

Comparison with Other Methods. We also compared our method with two of the best-performing story creation applications currently on the market: Artflow [14] and ComicAI [15], with results presented in Table 1. The results indicate that our method performed well on all these dimensions. Our scripts maintained the highest level of fidelity and logical coherence, even while our method allowing for the appropriate expansion of simple stories. This indicates that our script achieved a good narrative effect. For completeness, Artflow [14] and our AesopAgent are both good, but as we need to consider the final effect of the video, for generating narration, we require that the content shown in the image cannot be repeatedly described in narration, so we will lose some points when we do the script evaluation (only consider the text part), this is as expected. Overall our AesopAgent showed a significant improvement in the overall score and effect compared to other methods at the script generation stage. Professional screenwriters gave a high evaluation of our overall scripts and the final video.

4.1.3 Utilities Optimization

In this section, we will take the use of image generation utilities as examples to demonstrate the effect of our agent technology on the design of utilities and the optimization of utilities usage.

Utility Suggestion & Creation.

Leveraging agent technology has enabled us to effectively select and employ utilities that meet specific design requirements. We take the need for style consistency in the images of our stories as a case. As depicted in Figure 9, upon consulting our agent about the availability of utilities within the K-RAG framework that could satisfy our requirements, the agent analyzed and identified DALL-E 3 as the closest match. However, DALL-E 3 was found incapable of ensuring style consistency. As a result, the agent conducted an online search and proposed a solution combining Lora and ControlNet that could potentially achieve the desired style uniformity. Following a thorough evaluation by our experts, this proposed method was confirmed to be effective, culminating in the incorporation of both Lora and ControlNet into the K-RAG database.

Utilities Usage Optimization. Similar to the optimization process for prompts described in Figure 8, we utilized E-RAG techniques to refine the usage of our image generation utilities. To generate images that align with the requirements of our narratives, we imposed certain constraints on the usage of these utilities. In practice, we sometimes produced images that did not fulfill the video criteria. Leveraging the insights of domain experts and the feedback from GPT-4V, we refined the image descriptions, which significantly decreased the likelihood of such discrepancies. As depicted in Figure 10, in case 1, we occasionally produced images lacking backgrounds or with inappropriate framing. Experts and

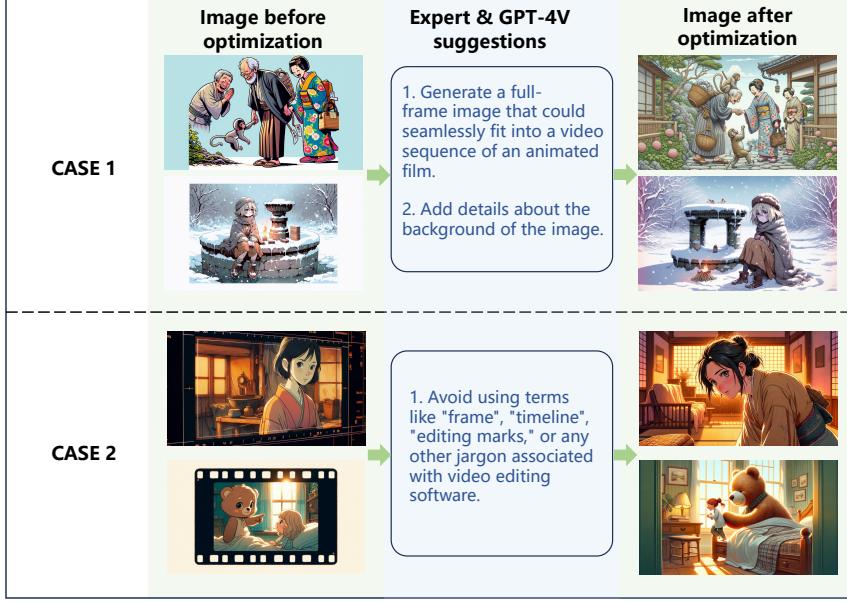


Figure 10: **Cases of utility usage optimization using E-RAG.** We present two cases of the use of E-RAG learning experts and GPT-4V experience to improve the quality of image generation.

GPT-4V suggest adding more detailed background descriptions and incorporating framing restrictions, the quality of the resulting images was noticeably enhanced. Case 2 illustrates the generation of images resembling film strips or appearing as if they were in the editing phase. By avoiding certain terminologies as suggested by experts and GPT-4V, the probability of generating this type of image is greatly reduced.

4.2 Main Results of Utility Layer

Here we show the results of the Utility Layer, including the case analysis and user study. We compare the generated images of our AesopAgent with other methods (i.e., SDXL [41] and ComicAI [15]) both qualitative and quantitative to show the effectiveness of the Utility Layer.

4.2.1 Effectiveness of Utility Modules

This experiment demonstrates the performance of our utilities. As shown in Table 2, we select image descriptions from two stories (i.e., “Goldilocks” and “Epaminondas and Auntie”) to explore the generated images from different methods. The generated images are displayed in Figure 11 where SDXL [41] and ComicAI [15] are employed as baselines. Here “Goldilocks” shows the blondie in various backgrounds, while “Epaminondas and Auntie” illustrates the interactions between two characters (i.e., a boy and his auntie).

Our utilities include three modules specifically, i.e., the image composition and character description refining module, the character inpainting module, and the image style consistency module. The image composition and character description refining module manages scene layout and character attributes. As indicated in Image 4, the interaction between the “Epaminondas” and “Auntie” is depicted accurately and vividly according to the given image description. The Character Inpainting module enhances character portrayal. For example, the “Goldilocks” in Image 1 and the “Epaminondas” in Image 5 are inpainted to change their skin tones to keep their image consistent. The image style consistency module effectively rendered the overall scene’s aesthetic. Contrasted to SDXL [41] and ComicAI [15], our utilities exhibit enhanced control in style and multi-character consistency. For instance, in Image 4, SDXL [41] draw the character “Auntie” as a boy, and ComicAI [15] ignores the character “Auntie”, while our utilities can control the consistency of the characters (i.e., “Epaminondas” and “Auntie”) and reserve the interaction among them. With the image style consistency module, we can render the images in different styles as shown in Figure 11. ComicAI [15] could

| Index | Image Description |
|----------------|---|
| Image 1 | A young girl with long golden curls on her face, her eyes wide and full of wonder. Wearing a simple blue and white ruffled dress, she has a mischievous look in her twinkling blue eyes, accentuated by her small, upturned nose. Little girl standing in front of an antique wooden house. |
| Image 2 | In the heart of a quaint kitchen filled with rustic wooden furniture and the aroma of breakfast, a young girl with long, curly golden hair and twinkling blue eyes is captured in a moment of anticipation. Her attire is a soft blue and white dress, ruffled from play. The girl holds a spoon precariously in her petite hand, raising it to her mouth as she gently blows on the steaming porridge scooped from the smallest bowl on the table. |
| Image 3 | Before an arrangement of three different-sized beds—each described as grand, moderate, and petite—stands a young girl with long, curly golden hair that cascades down her back, her shoulders drooped in exhaustion. In front of the broad first bed with a firm mattress, the girl presses her hands down on the bedding, her expression weary with a slight frown and a glint of hope in her twinkling blue eyes. |
| Image 4 | In a cozy, warm-toned kitchen, a young boy, possibly six or seven years old with short, curly, and unkempt hair and dressed in a slightly oversized soft shade of blue shirt and shorts that hit just above the knees, exhibits a kind, innocent expression. Eyes sparkling with wonder, the boy reaches out with his small hands, exploring a golden cake with gentle curiosity. Presenting the cake is a middle-aged woman with a benevolent smile, her hands showing signs of hard work, and wearing a dress with a full-length apron splattered with flour. |
| Image 5 | Morning sunlight streams through a window into a quaint rustic kitchen, illuminating the earthenware pots and fragrant herbs lining the shelves with a warm, amber glow. A young boy with short, curly hair and a kind expression stands at the center. The boy is about six or seven years old, dressed in a simple, oversized blue shirt and shorts, with well-worn shoes. A straw hat, with leaves peeking out, sits precariously atop his head, and a pound of butter, wrapped in leaves, is visible underneath the hat. A middle-aged woman with fine lines on her face, her hair held back by a cloth headband, stands to the side, her face showing a blend of affection and apprehension. |
| Image 6 | Under the relentless sun reigning high in the azure expanse, light pours radiance over the countryside. In the sweltering heat, a young boy around six or seven years old, with an innocent expression, short curly hair, dressed in a slightly oversized soft blue shirt, shorts just above the knees, and well-worn shoes, makes his way across the landscape. The golden cake that once resided in the young boy's hand now crumbles amid the overpowering warmth. |

Table 2: **The selected image descriptions for generated image comparison.** We select six image descriptions from two stories. The Image 1 to 3 are from the story “Goldilocks” and the Image 4 to 6 are from the story “Epaminondas and Auntie”.

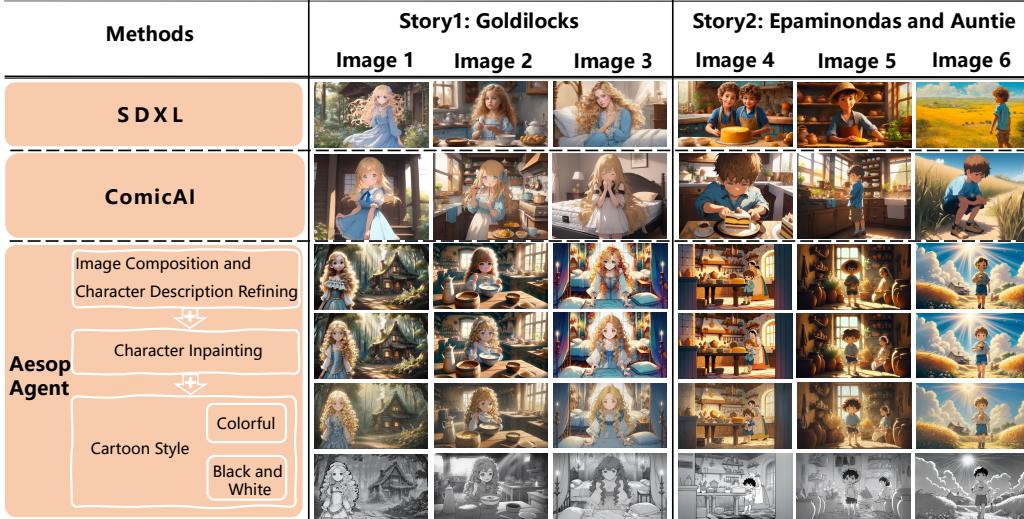


Figure 11: **Qualitative results of different methods on image generation.** We show the generated images from two stories (i.e., “Goldilocks” and “Epaminondas and Auntie”) from SDXL [41], ComicAI [15], and our AesopAgent. Specifically, the generated images through different AesopAgent’s utility modules are shown in Columns 3 to 6.

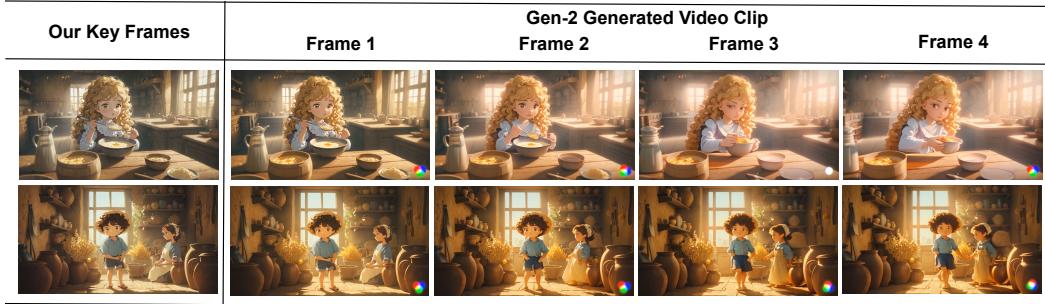


Figure 12: **Gen-2 [9] generated video clip according to our generated keyframes.** We use a keyframe to generate a 4-second video clip, and show the frames of the video clip.

| Methods | Fidelity (50%) | Rationality (30%) | Element State(20%) | Overall |
|-------------------|----------------|-------------------|--------------------|--------------|
| ComicAI [15] | 53.61 | 81.95 | 93.54 | 70.10 |
| AesopAgent (Ours) | 71.56 | 86.08 | 90.67 | 79.74 |

Table 3: **The results of the user study on generated image quality.** We collect more than 200 images from 10 stories, and compare their quality on fidelity, rationality, and Element State.

only use colorful cartoon style, and SDXL [41] does not focus on any certain style. Consequently, our Utility Layer achieves the industry standard, ensuring compositional coherence, consistent character depiction, and stylistic uniformity in image generation.

4.2.2 The Extension of Utilities

Our system could integrate additional units as utilities for enhancing storytelling. Here we employ an animating unit, i.e., Runway Gen-2 [9], to transform each generated keyframe into a 4-second video clip, as depicted in Figure 12. With Gen-2 [9], shots such as “Goldilocks” consuming porridge and the interaction in “Epaminondas and Auntie” are dynamically enriched, contributing to an immersive storytelling experience. So it is feasible for other utilities, such as Gen-2 [9] and Sora [40], to serve as downstream applications of our system.

4.3 Comparison with Other Methods

Here we compare our AesopAgent with other methods. We analyze the user study with ComicAI [15] and compare the storytelling ability with Human Design [16], NUWA-XL [64], and AutoStory [59].

4.3.1 User Study with ComicAI

We evaluate the image quality produced by two methods, (i.e., AesopAgent and ComicAI [15]). The qualitative results of ComicAI [15] have been shown in Figure 11, here we analyze the user study. Specifically, we randomly select 10 stories to produce above 200 shot images, then conduct manual scoring for each shot image. The evaluation is conducted on three dimensions: fidelity, rationality, and element state. Fidelity measures how closely the elements in the generated images correspond to those in the scripts, with deductions made for missing or misaligned basic elements. Rationality assesses the social realism of depicted elements, deeming elements that are non-existent in reality would be considered irrational (e.g., half-horse). Element State mainly analyzes spatial relationships between visual elements and their environment.

The average scores of these dimensions across the 10 stories are listed in Table 3 where these images are given by five annotators. To determine the overall image quality, we employed a weighted sum, assigning weights of 50%, 30%, and 20% to fidelity, rationality, and element state, respectively. Our AesopAgent obtains a slightly lower score on element state since our generated images contain more elements, making it difficult to organize every element precisely. The results also indicate that AesopAgent surpasses ComicAI [15] by 17.95 points in fidelity and by 4.13 points in rationality. This superior performance is credited to our Utility Layer. In detail, the image composition and

| Stories | Methods | Selected Key Frames | | | |
|---------------------|-------------------|---|--|---|--|
| | | Frame 1 | Frame 2 | Frame 3 | Frame 4 |
| Tang Poems | Human Design | The child poles a boat | Coming back secretly with lotus white | Knowing not how to hide tracks | Since trace of duckweed leaves |
| | AesopAgent (Ours) |  |  |  |  |
| Flintstones | NUWA-XL | Fred is walking in the room. | Fred and Wilma are talking in the room. | Fred and Betty are saying something on the road. | Fred is driving a red car on the road. |
| | AesopAgent (Ours) |  |  |  |  |
| The Cat and the Dog | AutoStory | The cat took the dog to a clear river nearby. They swam in the river together. The cool river water made them feel refreshed. | The dog took out his favorite toy, a red ball, and shared it with his new friend, the cat. They had a great time together. | The dog and cat went to a flat grassland, where they could play ball to their heart's content. Both of them were very excited to meet a new friend. | The dog and cat enjoyed a happy afternoon and were exhausted. They took a sunlit nap together. Basked in the sun, they know their warm friendship would never end. |
| | AesopAgent (Ours) |  |  |  |  |

Figure 13: **Qualitative comparison with other methods.** We compare the keyframes of our AesopAgent with the other three methods (i.e., Human Design [16], NUWA-XL [64], and AutoStory [59]) on three stories. The semantic meaning of each frame is listed above the corresponding generated image.

character description refining module visually represents key script elements; the character inpainting module inpaints the details within these elements; and the image style consistency module ensures the cohesion of the overall style. This structured approach allows our generated images to accurately reflect the script requirements, thereby outperforming ComicAI [15] in the overall score.

4.3.2 Qualitatively Comparison

Here we compare our method with other methods. As shown in Figure 13, our method compares with Human Design [16], NUWA-XL [64], and AutoStory [59] in three different stories: “Tang Poems”, “Flintstones”, and “The Cat and the Dog”. In the “Tang Poems”, Human Design tends to use simplistic visuals to convey the story’s content; while our method would render more detailed and aesthetically pleasing scenes. For the story “Flintstones”, it is obvious that our method excels at

generating the shot with actions (e.g., walking, dialogues involving multiple characters, and driving a car), all of which are more vivid compared with NUWA-XL [64]. In the story “The Cat and the Dog”, our method successfully captures the playfulness and friendship between the cat and dog, whereas AutoStory [59] simply places the cat and the dog in the same frame without capturing their interaction.

5 Conclusion

We propose AesopAgent, an agent system capable of converting user story proposals into videos. This system consists of two primary components: the Horizontal Layer and the Utility Layer. In the Horizontal Layer, we iteratively construct E-RAG and K-RAG based on expert experience and professional knowledge, leading a *RAG-based evolutionary system*. This approach facilitates efficient *task workflow orchestration*, prompt optimization, and utilities usage. The Utility Layer focuses on providing utilities to ensure *consistent image generation* through image composition rationality, consistency across multiple characters, and image style consistency. Furthermore, AesopAgent assembles images, audio, and narration into videos enhanced with special effects. Experiments demonstrate that it is feasible to use agent techniques to optimize workflow design and utilities usage for complex tasks like video generation. The generated scripts and images of our AesopAgent obtain higher scores than other similar systems, such as ComicAI [15] and Artflow [14]. And its overall narrative capability surpasses the previous research, including NUWA-XL [64] and AutoStory [59]. Our system can be further integrated with additional downstream AI utilitise (e.g., Gen-2 [9] and Sora [40]) to meet the requirements of individual users. Our future work aims to meet a broader range of user requirements, such as scripts of various themes and genres, a greater diversity of styles, and user-defined characters. This endeavor could even extend to assist the movie generation. At the same time, we will develop a more intelligent agent system, which can better understand the user’s intention and complete the user’s complex tasks quickly and conveniently.

References

- [1] Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse H. Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matthew Sharifi, Neil Zeghidour, and Christian Havnø Frank. MusicLM: Generating music from text. *CoRR*, abs/2301.11325, 2023.
- [2] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions.
- [3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.
- [4] Hong Chen, Rujun Han, Te-Lin Wu, Hideki Nakayama, and Nanyun Peng. Character-centric story visualization via visual planning and token alignment. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 8259–8272. Association for Computational Linguistics, 2022.
- [5] Xin Cheng, Di Luo, Xiuying Chen, Lemao Liu, Dongyan Zhao, and Rui Yan. Lift yourself up: Retrieval-augmented text generation with self memory, 2023.
- [6] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

- [7] Ming Ding, Wendi Zheng, Wenyi Hong, and Jie Tang. Cogview2: Faster and better text-to-image generation via hierarchical transformers. *Advances in Neural Information Processing Systems*, 35:16890–16902, 2022.
- [8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.
- [9] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7312–7322, 2023.
- [10] Zhangyin Feng, Xiaocheng Feng, Dezhong Zhao, Maojin Yang, and Bing Qin. Retrieval-generation synergy augmented large language models. *arXiv preprint arXiv:2310.05149*, 2023.
- [11] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.
- [12] Yingqiang Ge, Wenyue Hua, Kai Mei, Jianchao Ji, Juntao Tan, Shuyuan Xu, Zelong Li, and Yongfeng Zhang. Openagi: When llm meets domain experts, 2023.
- [13] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Neural Information Processing Systems*, 2014.
- [14] ArtFlow Group. Artflow. <https://app.artflow.ai/>, 2023. [online; 2024-03].
- [15] ComicAI Group. Transform your story into comics with ai image generation. <https://comicai.ai/dashboard>, 2023. [online; 2024-01].
- [16] Hankan. human design. <https://haokan.baidu.com/v?pd=wisenatural&vid=7320413949705196540>, 2023. [online; 2024-01].
- [17] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.
- [18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020.
- [19] Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, and Chenglin Wu. Metagpt: Meta programming for multi-agent collaborative framework. *CoRR*, abs/2308.00352, 2023.
- [20] Susung Hong, Junyoung Seo, Sunghwan Hong, Heeseong Shin, and Seungryong Kim. Large language models are frame-level directors for zero-shot text-to-video generation. *CoRR*, abs/2305.14330, 2023.
- [21] Chenxu Hu, Jie Fu, Chenzhuang Du, Simian Luo, Junbo Zhao, and Hang Zhao. Chatdb: Augmenting llms with databases as their symbolic memory. *CoRR*, abs/2306.03901, 2023.
- [22] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.
- [23] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer: Generating music with long-term structure. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

- [24] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qian-glong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*, 2023.
- [25] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2016.
- [26] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.
- [27] Pika Labs. Pika. <https://pika.art/>, 2023. [online; 2024-01].
- [28] Bowen Li. Word-level fine-grained story visualization. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXVI*, volume 13696 of *Lecture Notes in Computer Science*, pages 347–362. Springer, 2022.
- [29] Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbulin, and Bernard Ghanem. CAMEL: communicative agents for "mind" exploration of large scale language model society. *CoRR*, abs/2303.17760, 2023.
- [30] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR, 2023.
- [31] Yitong Li, Zhe Gan, Yelong Shen, Jingjing Liu, Yu Cheng, Yuexin Wu, Lawrence Carin, David E. Carlson, and Jianfeng Gao. Storygan: A sequential conditional GAN for story visualization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6329–6338. Computer Vision Foundation / IEEE, 2019.
- [32] Yitong Li, Martin Renqiang Min, Dinghan Shen, David Carlson, and Lawrence Carin. Video generation from text. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI'18/IAAI'18/EAAI'18*. AAAI Press, 2018.
- [33] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *CoRR*, abs/2304.08485, 2023.
- [34] Jerry Liu. LlamaIndex. https://github.com/jerryjliu/llama_index, 11 2022.
- [35] Fuchen Long, Zhaofan Qiu, Ting Yao, and Tao Mei. Videodrafter: Content-consistent multi-scene video generation with LLM. *CoRR*, abs/2401.01256, 2024.
- [36] Adyasha Maharana and Mohit Bansal. Integrating visuospatial, linguistic, and commonsense structure into story visualization. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6772–6786. Association for Computational Linguistics, 2021.
- [37] Ali Modarresi, Ayyoob Imani, Mohsen Fayyaz, and Hinrich Schütze. RET-LLM: towards a general read-write memory for large language models. *CoRR*, abs/2305.14322, 2023.
- [38] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [39] OpenAI. Gpt-4v(ision). https://cdn.openai.com/papers/GPTV_System_Card.pdf, 2023.

- [40] Bill Peebles, Tim Brooks, Connor Holmes, Clarence Ng, David Schnurr Eric Luhman, Joe Taylor, Li Jing, Natalie Summers, Ricky Wang, Rohan Sahai, Ryan O’Rourke, Troy Luhman, Will DePue, and Yufei Guo. Openai sora. <https://openai.com/research/video-generation-models-as-world-simulators>, 2024. [online; 2024-03].
- [41] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: improving latent diffusion models for high-resolution image synthesis. *CoRR*, abs/2307.01952, 2023.
- [42] Chen Qian, Xin Cong, Cheng Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong Sun. Communicative agents for software development. *CoRR*, abs/2307.07924, 2023.
- [43] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Xaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. Toolllm: Facilitating large language models to master 16000+ real-world apis, 2023.
- [44] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghuandan H Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q Tran, Jonah Samost, et al. Recommender systems with generative retrieval. *arXiv preprint arXiv:2305.05065*, 2023.
- [45] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International conference on machine learning*, pages 1060–1069. PMLR, 2016.
- [46] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2022.
- [47] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells III, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer, 2015.
- [48] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 22500–22510. IEEE, 2023.
- [49] Shikhar Sharma, Dendi Suhubdy, Vincent Michalski, Samira Ebrahimi Kahou, and Yoshua Bengio. Chatpainter: Improving text to image generation using dialogue. *arXiv preprint arXiv:1802.08216*, 2018.
- [50] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.
- [51] Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 11854–11864. IEEE, 2023.
- [52] S. Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1526–1535, 2017.
- [53] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6306–6315, 2017.

- [54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.
- [55] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual descriptions. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.
- [56] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Neural Information Processing Systems*, 2016.
- [57] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *CoRR*, abs/2305.16291, 2023.
- [58] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023.
- [59] Wen Wang, Canyu Zhao, Hao Chen, Zhekai Chen, Kecheng Zheng, and Chunhua Shen. Autostory: Generating diverse storytelling images with minimal human effort. *CoRR*, abs/2311.11243, 2023.
- [60] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. Autogen: Enabling next-gen LLM applications via multi-agent conversation framework. *CoRR*, abs/2308.08155, 2023.
- [61] Qiaolin Xia, Haoyang Huang, Nan Duan, Dongdong Zhang, Lei Ji, Zhifang Sui, Edward Cui, Taroon Bharti, and Ming Zhou. XGPT: cross-modal generative pre-training for image captioning. In Lu Wang, Yansong Feng, Yu Hong, and Ruifang He, editors, *Natural Language Processing and Chinese Computing - 10th CCF International Conference, NLPCC 2021, Qingdao, China, October 13-17, 2021, Proceedings, Part I*, volume 13028 of *Lecture Notes in Computer Science*, pages 786–797. Springer, 2021.
- [62] Hui Yang, Sifu Yue, and Yunzhong He. Auto-gpt for online decision making: Benchmarks and additional opinions. *CoRR*, abs/2306.02224, 2023.
- [63] Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. MM-REACT: prompting chatgpt for multimodal reasoning and action. *CoRR*, abs/2303.11381, 2023.
- [64] Shengming Yin, Chenfei Wu, Huan Yang, Jianfeng Wang, Xiaodong Wang, Minheng Ni, Zhengyuan Yang, Linjie Li, Shuguang Liu, Fan Yang, et al. NUWA-XL: Diffusion over Diffusion for eXtremely Long Video Generation. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2023, Toronto, Canada, July 9-14, 2023, pages 1309–1320. Association for Computational Linguistics, 2023.
- [65] Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. Making retrieval-augmented language models robust to irrelevant context. *arXiv preprint arXiv:2310.01558*, 2023.
- [66] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *Trans. Mach. Learn. Res.*, 2022, 2022.
- [67] Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. Chain-of-note: Enhancing robustness in retrieval-augmented language models. *arXiv preprint arXiv:2311.09210*, 2023.

- [68] Zichun Yu, Chenyan Xiong, Shi Yu, and Zhiyuan Liu. Augmentation-adapted retriever improves generalization of language models as generic plug-in. *arXiv preprint arXiv:2305.17331*, 2023.
- [69] Ao Zhang, Wei Ji, and Tat-Seng Chua. Next-chat: An LMM for chat, detection and segmentation. *CoRR*, abs/2311.04498, 2023.
- [70] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3813–3824, 2023.
- [71] Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. Retrieve anything to augment large language models. *arXiv preprint arXiv:2310.07554*, 2023.
- [72] Guangcong Zheng, Xianpan Zhou, Xuewei Li, Zhongang Qi, Ying Shan, and Xi Li. Layoutdiffusion: Controllable diffusion model for layout-to-image generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 22490–22499. IEEE, 2023.
- [73] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.
- [74] Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, Yu Qiao, Zhaoxiang Zhang, and Jifeng Dai. Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory. *CoRR*, abs/2305.17144, 2023.
- [75] Shaobin Zhuang, Kunchang Li, Xinyuan Chen, Yaohui Wang, Ziwei Liu, Yu Qiao, and Yali Wang. Vlogger: Make your dream a vlog. *arXiv preprint arXiv:2401.09414*, 2024.