VideoStudio: Generating Consistent-Content and Multi-Scene Videos

Fuchen Long, Zhaofan Qiu, Ting Yao, and Tao Mei

HiDream.ai Inc.
{longfuchen, qiuzhaofan, tiyao, tmei}@hidream.ai

Abstract. The recent innovations and breakthroughs in diffusion models have significantly expanded the possibilities of generating high-quality videos for the given prompts. Most existing works tackle the single-scene scenario with only one video event occurring in a single background. Extending to generate multi-scene videos nevertheless is not trivial and necessitates to nicely manage the logic in between while preserving the consistent visual appearance of key content across video scenes. In this paper, we propose a novel framework, namely VideoStudio, for consistent-content and multi-scene video generation. Technically, VideoStudio leverages Large Language Models (LLM) to convert the input prompt into comprehensive multi-scene script that benefits from the logical knowledge learnt by LLM. The script for each scene includes a prompt describing the event, the foreground/background entities, as well as camera movement. VideoStudio identifies the common entities throughout the script and asks LLM to detail each entity. The resultant entity description is then fed into a text-to-image model to generate a reference image for each entity. Finally, VideoStudio outputs a multi-scene video by generating each scene video via a diffusion process that takes the reference images, the descriptive prompt of the event and camera movement into account. The diffusion model incorporates the reference images as the condition and alignment to strengthen the content consistency of multi-scene videos. Extensive experiments demonstrate that VideoStudio outperforms the SOTA video generation models in terms of visual quality, content consistency, and user preference. Source code is available at `https://github.com/FuchenUSTC/VideoStudio`.

1 Introduction

Diffusion Probabilistic Models (DPM) have demonstrated high capability in generating high-quality images [7,15,16,35,36,44,48,50,68]. DPM approaches image generation as a multi-step sampling process, involving the use of a denoiser network to progressively transform a Gaussian noise map into an output image. Compared to 2D images, videos have an additional time dimension, which introduces more challenges when extending DPM to video domain. One typical way is to leverage pre-trained text-to-image models to produce video frames [20,40,59]

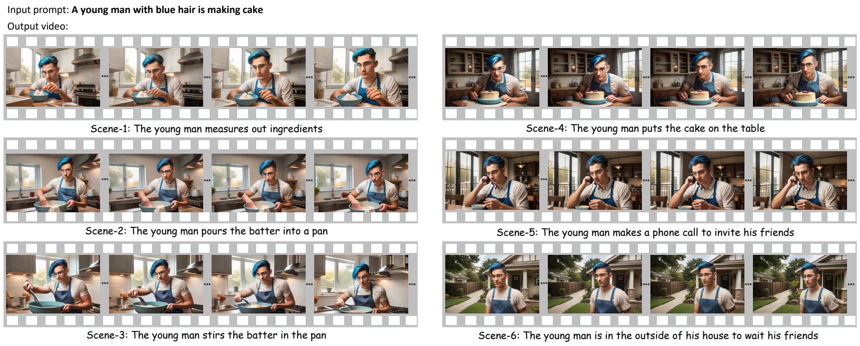

Input prompt: **A young man with blue hair is making cake**
Output video:
Scene-1: The young man measures out ingredients
Scene-2: The young man pours the batter into a pan
Scene-3: The young man stirs the batter in the pan
Scene-4: The young man puts the cake on the table
Scene-5: The young man makes a phone call to invite his friends
Scene-6: The young man is in the outside of his house to wait his friends

Fig. 1: An illustration of prompt and multi-scene video generation by VideoStudio.

or utilize a 3D denoiser network learnt on video data to generate a sequence of frames in an end-to-end manner [3, 11, 12, 14, 34, 47]. Despite having impressive results in the realm of text-to-video generation, most existing works focus on only single-scene videos, featuring one event in a single background. The generation of multi-scene video is still a problem not yet fully explored in the literature.

The difficulty of multi-scene video generation generally originates from two aspects: 1) how to arrange and establish different events in a logical and realistic way for a multi-scene video? 2) how to guarantee the consistency of common entities, e.g., foreground objects or persons, throughout the video? For instance, given an input prompt of "a young man is making cake," a multi-scene video is usually to present the step-by-step procedure of making a cake, including measuring out the ingredients, pouring the ingredients into a pan, cooking the cake, etc. This necessitates a comprehensive understanding and refinement of the prompt. As such, we propose to mitigate the first issue through capitalizing on Large Language Models (LLM) to rewrite the input prompt into multi-scene video script. LLM inherently abstracts quantities of text data on the Web about the input prompt to produce the script, which describes and decomposes the video logically into multiple scenes. To alleviate the second issue, we exploit the common entities to generate reference images as the additional condition to produce each scene video. The reference images, as the link across scenes, effectively align the content consistency within a multi-scene video.

To consolidate the idea, we present a new framework dubbed as **VideoStudio** for consistent-content and multi-scene video generation. Technically, VideoStudio first transforms the input prompt into a thorough multi-scene video script by using LLM. The script for each scene consists of the descriptive prompt of the event in the scene, a list of foreground objects or persons, the background, and camera movement. VideoStudio then identifies common entities that appear across multiple scenes and requests LLM to enrich each entity. The resultant entity description is fed into a pre-trained Stable Diffusion [44] model to produce a reference image for each entity. Finally, VideoStudio outputs a multi-scene video via involving two diffusion models, i.e., **VideoStudio-Img** and **VideoStudio-**

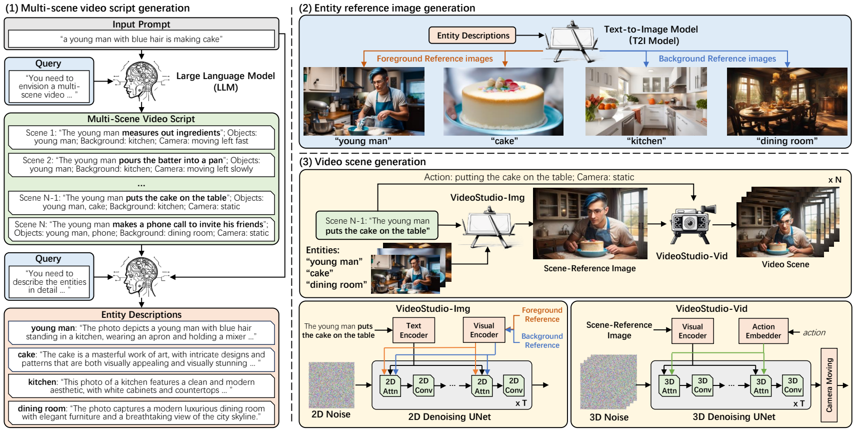
Vid. VideoStudio-Img is dedicated to incorporating the descriptive prompt of the event and the reference images of entities in each scene as the condition to generate a scene-reference image. VideoStudio-Vid takes the scene-reference image plus temporal dynamics of the action depicted in the descriptive prompt of the event and camera movement in the script as the inputs and produces a video clip for each scene.

The main contribution of this work is the proposal of VideoStudio for generating consistent-content and multi-scene videos. The solution also leads to the elegant views of how to use LLM to properly arrange content of multi-scene videos and how to generate visually consistent entities across scenes, which are problems seldom investigated in literature. Extensive experiments conducted on public benchmarks demonstrate that VideoStudio outperforms SOTA video generation models in terms of visual quality, content consistency and user preference.

2 Related Work

Image generation is a fundamental challenge of computer vision and has evolved rapidly in the past decade. Recent advances in Diffusion Probabilistic Models (DPM) have led to remarkable improvements in generating high-fidelity images [3,7,15,16,32,33,35–37,43,44,48–50,68]. DPM is a category of generative models that utilizes a sequential sampling process to convert random Gaussian noise into high-quality images. For example, GLIDE [37] and DALL-E 2 [43] exploit the sampling process in the pixel space, conditioned on the text prompt using classifier-free guidance [16]. Nevertheless, training a powerful denoising network remains challenging due to high computational cost and memory demand associated with sampling at the pixel level. To mitigate this problem, Latent Diffusion Models (LDM) [44] employ sampling in the latent feature space that is established by a pre-trained autoencoder, leading to the improvements on computation efficiency and image quality. Furthermore, the application of DPM is further enhanced by incorporating advanced sampling strategies [32, 33, 49] and additional control signals [35, 68].

Video generation is a natural extension of image generation in video domain. The early approaches, e.g., ImagenVideo [14] and Make-A-Video [47], train video diffusion models in the pixel space, resulting in high computational complexity. Following LDM in image domain, several works [3, 6, 11, 34, 70] propose to exploit the sampling process in the latent feature space for video generation. These works extend the 2D UNet with the transformer layers [23, 62, 63] to 3D UNet by injecting temporal self-attentions [28, 29] and/or temporal convolutions [30,31]. For instance, Video LDM [3] and AnimateDiff [11] focus on training the injected temporal layers while freezing the spatial layers to preserve the ability of the pre-trained image diffusion model. VideoFusion [34] decomposes the 3D noise into a 2D base noise shared across frames and a 3D residual noise, enhancing the correlation between frames. However, the generated videos usually have a limited time duration, typically around 16 frames. Consequently, some recent researches emerge to generate long videos by an extrapolation strategy or



"a young man with blue hair is making cake"

"You need to envision a multi-scene video ... "

Scene 1: "The young man **measures out ingredients**"; Objects: young man; Background: kitchen; Camera: moving left fast

Scene 2: "The young man **pours the batter into a pan**"; Objects: young man; Background: kitchen; Camera: moving left slowly

Scene N-1: "The young man **puts the cake on the table**"; Objects: young man, cake; Background: kitchen; Camera: static

Scene N: "The young man **makes a phone call to invite his friends**"; Objects: young man, phone; Background: dining room; Camera: static

"You need to describe the entities in detail ... "

young man: "The photo depicts a young man with blue hair standing in a kitchen, wearing an apron and holding a mixer ..."

cake: "The cake is a masterful work of art, with intricate designs and patterns that are both visually appealing and visually stunning ... "

kitchen: "This photo of a kitchen features a clean and modern aesthetic, with white cabinets and countertops ... "

dining room: "The photo captures a modern luxurious dining room with elegant furniture and a breathtaking view of the city skyline."

Scene N-1: "The young man **puts the cake on the table**"

The young man **puts the cake on the table**

Fig. 2: An overview of our VideoStudio framework for consistent-content and multi-scene video generation. VideoStudio consists of three main stages: (1) multi-scene video script generation, (2) entity reference image generation, and (3) video scene generation. In the first stage, LLM is utilized to convert the input prompt into a comprehensive multi-scene script. The script for each scene includes the descriptive prompt of the event in the scene, a list of foreground objects or persons, the background, and camera movement. We then request LLM to detail the common foreground/background entities across scenes. These entity descriptions are fed into a text-to-image (T2I) model to produce reference images in the second stage. Finally, in the third stage, VideoStudio-Img exploits the descriptive prompt of the event and the reference images of entities in each scene as the condition to generate a scene-reference image. VideoStudio-Vid takes the scene-reference image plus temporal dynamics of the action depicted in the descriptive prompt of the event and camera movement in the script as the inputs and produces a video clip for each scene.

hierarchical architecture [12, 25, 52, 53, 66]. In addition, video editing techniques utilize the input video as a condition and generate a video by modifying the style or key object of the input video [9, 10, 12, 18, 39, 40, 46, 54, 57, 59, 65].

In short, our work in this paper focuses on consistent-content and multi-scene video generation. The most related work is [26], which aligns the appearance of entities across scenes through the bounding boxes provided by LLM. Ours is different in the way that we explicitly determine the appearance of entities by generating reference images, which serve as a link across scenes and effectively enhance the content consistency within a multi-scene video.

3 VideoStudio

This section presents the proposed VideoStudio framework for consistent-content and multi-scene video generation. Figure 2 illustrates an overview of VideoStudio framework, consisting of three main stages: (1) multi-scene video script generation (Sec. 3.1), (2) entity reference image generation (Sec. 3.2), and (3) video scene generation (Sec. 3.3).

3.1 Multi-Scene Video Script Generation

As depicted in Figure 2(1), VideoStudio utilizes LLM to convert the input prompt into a comprehensive multi-scene script. In view of its high deployment flexibility and inference efficiency, we use the open-source ChatGLM3-6B model [8, 67]. The LLM is requested by a pre-defined query, *"You need to envision a multi-scene video and describe each scene ..."*, to treat the input prompt as the theme, logically decompose the video into multiple scenes and generate a script for each scene in the following format:

$$\begin{aligned}
&\text{[Scene 1: prompt, foreground, background, camera move];}\\
&\text{[Scene 2: prompt, foreground, background, camera move];}\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\ldots\\
&\text{[Scene } N\text{: prompt, foreground, background, camera move].}
\end{aligned} \tag{1}$$

Here N denotes the number of video scenes, which is determined by the LLM. For each scene, the descriptive prompt of the event in the scene, a list of foreground objects or persons, the background, and camera movement are provided. The camera movement is restricted to a close-set of directions *{static, left, right, up, down, forward, backward}* and speeds *{slow, medium, fast}*.

Next, VideoStudio identifies the common entities, which include foreground objects or persons and background locations. To achieve this, we ask the LLM to assign the common object, person, or background the same name across scenes when generating the video script. Therefore, we strictly match the name of entities and discover the entities that appear in multiple scenes. To further improve the quality of the video script, we employ the capability of the LLM for multi-round dialogue. Specifically, we start the dialogue by asking the LLM to specify the key aspects with respect to the entity, such as *"What are the aspects that should be considered when describing a photo of a young man in detail?"* In the next round of dialogue, we request the LLM to describe the entity from the viewpoints of the given aspects. Moreover, the original prompt is also taken as the input to the LLM to ensure that the essential characteristics, e.g., "blue hair" of the young man, are emphasized in entity description generation.

Please note that the GPT-4 [38] can also be used for script generation, but it incurs an additional 0.12 USD for the GPT-4 API call per query. In VideoStudio, we leverage the open-source ChatGLM3-6B and perform the inference on our devices to circumvent the need for API call. Nevertheless, the scale of ChatGLM3-6B is much smaller, resulting in unstable outcomes that may deviate from the specified script format. To alleviate this issue, we have empirically abstracted the following principles to enhance the stability of open-source LLM:

- Before the dialogue starts, we provide comprehensive instructions to the LLM, delineating the additional requirements, specifying the script format, and offering the examples of the expected outputs.
- For each query, we manually select five in-context examples as the historical context for multi-round dialogue. These examples are very carefully designed to ensure a diverse range of scenes, key objects, and background, and serve to emphasize the required script format for LLM.

- After each round of dialogue, we verify the output format. If the results are seemingly inappropriate, we re-run the entire script generation stage. Such strategy is simple to implement without requiring any additional expenses.

We will provide the full version of our instructions, examples, and queries in the supplementary materials.

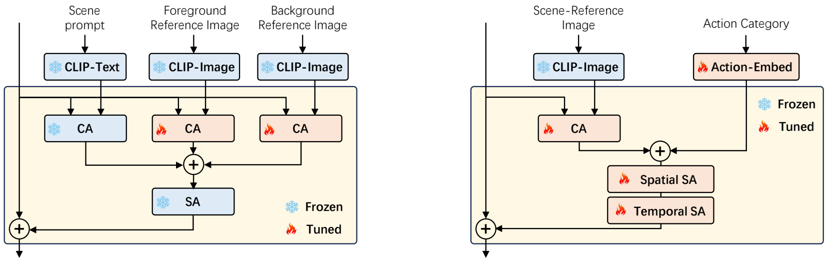
3.2 Entity Reference Image Generation

In the second stage of VideoStudio, we unify the visual appearance of common entities by explicitly generating a reference image for each entity. The reference images act as the link to cohere the content across scenes. We achieve this by first feeding the entity description into a pre-trained Stable Diffusion model for text-to-image generation. Then, we employ the U^2-Net [41] model for salient object detection, and segment the foreground and background areas in each resultant image. By utilizing the segmentation masks, we can further remove the background pixels from the foreground reference image and vice versa, in order to prevent the interference between the foreground and background visual contents in the reference images.

3.3 Video Scene Generation

VideoStudio produces a multi-scene video by generating each scene via the diffusion models by taking the reference images, the descriptive prompt of the event and camera movement into account. This stage involves two primary components: the **VideoStudio-Img**, which utilizes the descriptive prompt of the event and the reference images of entities in each scene as the condition to generate a scene-reference image, and the **VideoStudio-Vid**, which employs the scene-reference image plus temporal dynamics of the action depicted in the descriptive prompt of the event and camera movement in the script as the inputs and produces a video clip for each scene.

The **VideoStudio-Img** component aims to generate a scene-reference image conditioning on the event prompt and entity reference images for each scene. To accomplish this, we remold the Stable Diffusion architecture by replacing the original attention module with a novel attention module that can handle three contexts: the text prompt, foreground reference image, and background reference image. As depicted in Figure 3a, we utilize text and visual encoder of a pre-trained CLIP model to extract the sequential text feature $y_t \in \mathbb{R}^{L_t \times C_t}$ and local image features $y_f \in \mathbb{R}^{L_f \times C_f}$ and $y_b \in \mathbb{R}^{L_b \times C_b}$ for the prompt, foreground reference image, and background reference image, respectively. Here, L and C denote the length and the channels of the feature sequence. For the case of multiple foregrounds in one scene, we concatenate the features from all foreground reference images along the length dimension. Given the input feature $\boldsymbol{x}$, the



(a) **(b)**

Fig. 3: Diagram illustrations of (a) attention module in the VideoStudio-Img which takes the scene prompt and foreground/background reference images as the inputs and (b) attention module in the VideoStudio-Vid conditioning on the scene-reference image and the described action category.

outputs $\boldsymbol{z}$ of the attention are computed as

$$\begin{aligned}\boldsymbol{y} &= \mathrm{CA}_1(\boldsymbol{x}, y_t) + \mathrm{CA}_2(\boldsymbol{x}, y_f) + \mathrm{CA}_3(\boldsymbol{x}, y_b), \\ \boldsymbol{z} &= \boldsymbol{x} + \mathrm{SA}(\boldsymbol{y}),\end{aligned} \tag{2}$$

where CA_1 and SA are the cross-attention and self-attention modules, respectively, in the original Stable Diffusion architecture. We add two additional cross-attention modules, CA_2 and CA_3, which leverage the guidance provided by entity reference images. Moreover, we propose to optimize the parameters of CA_2 and CA_3 while freezing the other parts of the network.

The **VideoStudio-Vid** is a video diffusion model that employs the scene-reference image, the action described in the prompt of the event, and camera movement in the script as the inputs. Particularly, we start by extending the Stable Diffusion model to a spatio-temporal form and replacing the original attention module with a new one that is conditioned on the scene-reference image and action category, as shown in Figure 3b. Taking 400 action categories in Kinetics [4] as an action vocabulary, an indicator vector $y_a \in [0, 1]^{400}$ is built to infer if each action in the vocabulary exists in the scene prompt and subsequently converted into feature space using a linear embedding f. For the scene-reference image, we use the visual encoder of CLIP to extract the image feature $y_s \in \mathbb{R}^{L_s \times C_s}$, which is then fed into the cross-attention operation. The original self-attention is decomposed into a spatial self-attention (Spatial SA) and a temporal self-attention (Temporal SA), which operate self-attention solely on spatial and temporal dimension, respectively, to reduce computations. Hence, given the input feature $\boldsymbol{x}$, the attention module is formulated as

$$\begin{aligned}\boldsymbol{y} &= \mathrm{CA}(\boldsymbol{x}, y_s) + f(y_a), \\ \boldsymbol{z} &= \boldsymbol{x} + \text{Temporal SA}(\text{Spatial SA}(\boldsymbol{y})).\end{aligned} \tag{3}$$

Moreover, we further inject several temporal convolutions behind each spatial convolution into the Stable Diffusion architecture, to better capture temporal dependencies in image-to-video generation.

To reflect the camera movement stated by the script in the generated video, we uniquely modify the frames in the intermediate step of sampling process by warping the neighboring frames based on the camera moving direction and speed. We execute this adjustment after the first T_m DDIM sampling steps, followed by continuing the sampling process. Such modification ensures that the resultant video clip maintains the same camera movement as we warp the intermediate frames. In general, setting a small T_m for early modification may not effectively control the camera movement, while a late modification may affect the visual quality of the output videos. In practice, we observe that T_m=5 provides a good trade-off. We will detail the formulation of the modification process and the ablation study of the step T_m in our supplementary materials.

4 Experiments

4.1 Datasets

Our VideoStudio framework is trained on three large-scale datasets: LAION-2B [45], WebVid-10M [1] and HD-VG-130M [56]. The LAION-5B is one of the largest text-image dataset consisting of around 5 billion text-image pairs. To train VideoStudio-Img, We utilize a subset, namely **LAION-2B**, which focuses on the text prompts in English. The **WebVid-10M** and **HD-VG-130M** are the large-scale single-scene video datasets, containing approximately 10M and 130M text-video pairs, respectively. VideoStudio-Vid is trained on the combination of WebVid-10M and a randomly chosen 20M subset from HD-VG-130M.

To evaluate video generation, we select the text prompts from three video datasets, i.e., MSR-VTT [61], ActivityNet Captions [22] and Coref-SV [26]. The first one provides the single-scene prompts, while the remaining two datasets comprise multi-scene prompts. The **MSR-VTT** consists of 10K web video clips, each annotated with approximate 20 natural sentences. We utilize the text annotation of validation videos to serve as single-scene prompts in our evaluation. The **ActivityNet Captions** dataset is a multi-event video dataset designed for dense-captioning tasks. Following [26], we randomly sample 165 videos from the validation set and exploit the event captions as the multi-scene prompts. The **Coref-SV** is a multi-scene description dataset, which was constructed by replacing the subject of multi-scene paragraphs in Pororo-SV dataset [21,24]. Coref-SV samples 10 episodes from the Pororo-SV dataset and replaces the subject with 10 real-world entities, resulting in 100 multi-scene prompts.

4.2 Evaluation Metrics

For the video generation task, we adopt five evaluation metrics. To assess the visual quality of the generated videos, we utilize the average of the per-frame Fréchet Inception Distance (**FID**) [13] and the clip-level Fréchet Video Distance (**FVD**) [51], both of which are commonly used metrics. We also employ the **CLIPSIM** [58] metric to evaluate the alignment between the generated frames and the input prompt. To verify the content consistency, we calculate frame consistency (**Frame Consis.**) by determining the CLIP-similarity between consecutive frames, serving as an intra-scene consistency measure. Additionally, we employ the Grounding-DINO detector [27] to detect common objects across scenes and then calculate the CLIP-similarity between the common objects appeared in different scenes, achieving cross-scene consistency (**Scene Consis.**).

4.3 Implementation Details

We implement the proposed VideoStudio using the Diffusers codebase on the platform of PyTorch.

Training stage of VideoStudio-Img. VideoStudio-Img is originated from the Stable Diffusion v2.1 model by incorporating two additional cross-attention modules. These modules are initialized from scratch and trained on the text-image pairs from LAION-2B dataset, while other parts of the network are frozen. For each image, we randomly sample a 512×512 patch cropped from the original image, and utilize the U^2-Net model to segment the foreground area of each patch. The isolated foreground and background areas serve as the foreground and background reference images, respectively, to guide the generation of the input patch. We set each minibatch as 512 patches that are processed on 64 A100 GPUs in parallel. The parameters of the model are optimized by AdamW optimizer with a fixed learning rate of 1×10^{-4} for 20K iterations.

Training stage of VideoStudio-Vid. VideoStudio-Vid is developed based on the Stable Diffusion XL architecture by inserting temporal attentions and temporal convolutions. The training is carried out on the WebVid-10M and HD-VG-130M datasets. For each video, we randomly sample a 16-frame clip with the resolution of 320×512 and an FPS of 8. The middle frame of the clip is utilized as the scene-reference image. Each minibatch consists of 128 video clips implemented on 64 A100 GPUs in parallel. We utilize the AdamW optimizer with a fixed learning rate of 3×10^{-6} for 480K iterations.

4.4 Experimental Analysis of VideoStudio

Evaluation on VideoStudio-Img. We first verify the efficacy of VideoStudio-Img in aligning with the input entity reference images. To this end, we take the prompts from MSR-VTT validation set. The input foreground and background reference images are produced by using LLM and Stable Diffusion model. We validate the generated images on the measure of foreground similarity (**FG-SIM**) and background similarity (**BG-SIM**), which are the CLIP-similarity values with the foreground and background reference images, respectively. Table 1 lists the performance comparisons of IP-Adapter [64] and different VideoStudio-Img variants by leveraging different input references. Specifically, the use of foreground/background reference image as guidance leads to higher FG-SIM/BG-SIM values comparing to IP-Adapter or not leveraging reference images. Though both of IP-Adapter and VideoStudio-Img exploit additional cross-attention to maintain visual contents in image diffusion, our VideoStudio-Img is devised for

Table 1: Performance comparisons of IP-Adapter [64] and VideoStudio-Img variants with different input references on the MSR-VTT validation set.

Input References		**FG-SIM**	**BG-SIM**	**CLIPSIM**
FG Ref.	BG Ref.			
w/o Ref.		0.5162	0.4131	0.3001
IP-Adapter [64]				
✓		0.7116	0.4035	0.2910
	✓	0.5128	0.5059	0.2954
VideoStudio-Img				
✓		0.7919	0.4393	0.2982
	✓	0.5362	0.5742	0.3002
✓	✓	**0.8102**	**0.5861**	**0.3023**

Fig. 4: Examples of the foreground and background reference images and the generated scene-reference image by the VideoStudio-Img variants.

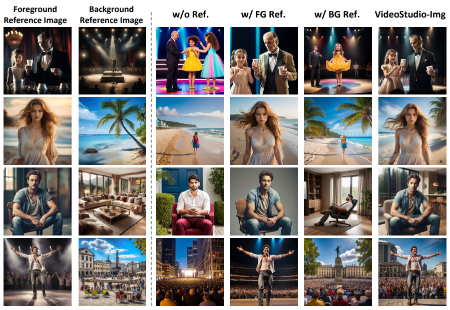


Table 2: Performance comparisons for single-scene video generation with real frame as scene-reference image on WebVid-10M.

Approach	**FVD** (↓)	**Frame Consis.** (↑)
RF+VideoCrafter [5]	293.3	97.9
RF+I2VGen-XL [69]	254.9	97.6
RF+VideoComposer [57]	231.0	95.9
RF+DynamiCrafter [60]	176.8	97.5
RF+SVD [2]	153.0	98.7
RF+VideoStudio-Vid$^-$	157.3	98.5
RF+VideoStudio-Vid	**116.5**	**98.8**

a more complex scenario to specify foreground objects and background. There are two major differences: 1) We pre-segment the foreground/background of the reference images to avoid the visual content interference; 2) IP-Adapter extracts global image features from CLIP, while ours utilizes local image tokens from CLIP to improve spatial discrimination in local regions. As indicated by the results, emphasizing the feature learning of local region on the more clean (masked) foreground/background reference image does benefit the visual alignment. Furthermore, the combination of both reference images achieves the highest FG-SIM of 0.8102 and BG-SIM of 0.5861. Figure 4 showcases four generated images by different VideoStudio-Img variants with various reference images. The results demonstrate the advantage of VideoStudio-Img to align with the visual contents in the entity reference images.

Evaluation on VideoStudio-Vid. Next, we assess the visual quality of the single-scene videos generated by VideoStudio-Vid. We exploit the real frame from the WebVid-10M validation set as the scene-reference image irrespective of the generation quality, and produce a video using the corresponding text prompt, which is referred to as RF+VideoStudio-Vid. We compare our proposal with five image-to-video diffusion models and one variant of VideoStudio-Vid, i.e., RF+VideoStudio-Vid$^-$, which disables the action guidance in VideoStudio-Vid. Table 2 presents the performance comparisons for single-scene video gen-

Table 3: Performance comparisons for single-scene video generation on MSR-VTT validation set. RF indicates whether to utilize the real frame as the reference.

Approach	**RF**	**FID** (↓)	**FVD** (↓)
CogVideo [17]		23.6	-
MagicVideo [71]		-	998
Make-A-Video [47]		13.2	-
VideoComposer [57]		-	580
VideoDirectorGPT [26]		12.2	550
ModelScopeT2V [55]		**11.1**	550
SD+VideoStudio-Vid		11.9	**381**
RF+VideoCrafter [5]	✓	45.0	339
RF+I2VGen-XL [69]	✓	37.4	264
RF+VideoComposer [57]	✓	31.3	208
RF+DynamiCrafter [60]	✓	26.1	196
RF+SVD [2]	✓	15.3	172
RF+VideoStudio-Vid	✓	**10.8**	**133**

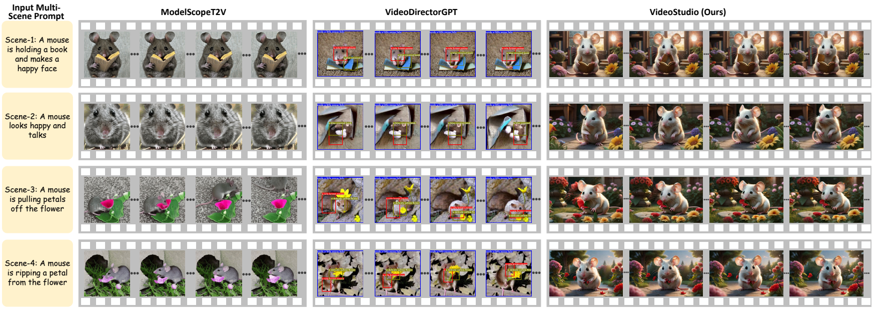

Input Multi-Scene Prompt
ModelScopeT2V
VideoDirectorGPT
VideoStudio (Ours)
Scene-1: A mouse is holding a book and makes a happy face
Scene-2: A mouse looks happy and talks
Scene-3: A mouse is pulling petals off the flower
Scene-4: A mouse is ripping a petal from the flower

Fig. 5: Examples of generated multi-scene videos by ModelScopeT2V [55], VideoDirectorGPT [26] and our VideoStudio utilizing a multi-scene prompt from the Coref-SV dataset. For each video, only the first four scenes are given. The results of VideoDirectorGPT are provided in the project webpage and thus with bounding box annotation.

eration on the WebVid-10M dataset. With the same scene-reference images, VideoStudio-Vid$^-$ outperforms most image-to-video approaches and obtains comparable FVD performance with the strong baseline SVD. The competitive result is attributed to the deep network architecture and large-scale training set. The performance is further enhanced to 116.5 FVD and 98.8 frame consistency by RF+VideoStudio-Vid, verifying the superiority of involving action category guidance to improve visual quality and intra-scene consistency.

Similar performance trends are observed on MSR-VTT dataset, as summarized in Table 3. The methods in this table are grouped into two categories: the methods with or without real frame (RF) as reference. To compare with the generation models without RF, we develop a two-step solution that first generates the scene-reference image by Stable Diffusion, and then converts the image into a video clip by VideoStudio-Vid, which is denoted as **SD+VideoStudio-Vid**. Specifically, VideoStudio-Vid attains the best FVD on both settings with and without a real frame as reference. SD+VideoStudio-Vid is slightly inferior to ModelScopeT2V in FID. We speculate that this may be the result of not optimizing Stable Diffusion on video frames, resulting in poorer frame quality against ModelScopeT2V. Nevertheless, SD+VideoStudio-Vid ap-

Table 4: Performance comparisons for multi-scene video generation on ActivityNet Captions dataset.

Approach	FID (↓)	FVD (↓)	Scene Consis. (↑)
ModelScopeT2V [55]	18.1	980	46.0
VideoDirectorGPT [26]	16.5	805	64.8
VideoStudio w/o Ref.	17.3	624	50.8
VideoStudio	**13.2**	**395**	**75.1**

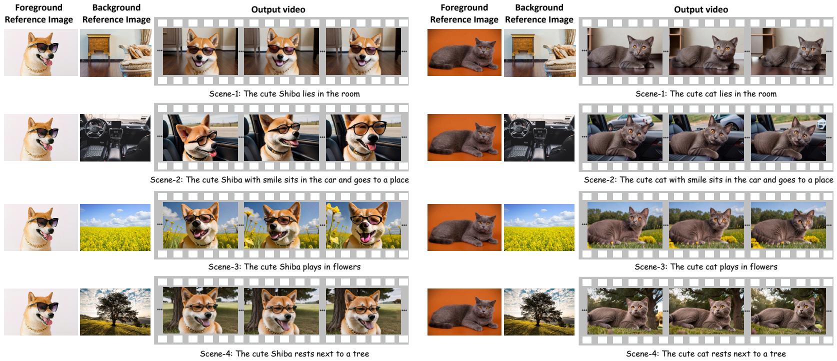

Foreground Reference Image
Background Reference Image
Output video
Scene-1: The cute Shiba lies in the room
Scene-2: The cute Shiba with smile sits in the car and goes to a place
Scene-3: The cute Shiba plays in flowers
Scene-4: The cute Shiba rests next to a tree
Foreground Reference Image
Background Reference Image
Output video
Scene-1: The cute cat lies in the room
Scene-2: The cute cat with smile sits in the car and goes to a place
Scene-3: The cute cat plays in flowers
Scene-4: The cute cat rests next to a tree

Fig. 6: Two examples of generated multi-scene videos by our VideoStudio using the real images as entity reference images.

parently surpasses ModelScopeT2V in FVD, validating the video-level quality by VideoStudio-Vid.

To evaluate the effectiveness of the action category condition for motion generation, we additionally implement an ablation study on the recent released VBench [19] benchmark. We measure the **action score** in VBench to assess whether human subjects can accurately execute the specific action mentioned in the text prompts. By using the action category as the condition in video diffusion, the action score of VideoStudio-Vid is improved from 90.3% to 96.5%, indicating the efficacy of action category condition to emphasize motion patterns.

4.5 Evaluations on Multi-Scene Video Generation

We validate VideoStudio for multi-scene video generation on ActivityNet Captions and Coref-SV datasets. Both of the datasets consist of multi-scene prompts, which necessitate the LLM to write the video script based on the given prompt of each scene. We compare with three approaches: ModelScopeT2V, VideoDirectorGPT and VideoStudio w/o Ref. by disabling the reference images in VideoStudio. Table 4 details the performance comparisons on ActivityNet Captions. As indicated by the results in the table, VideoStudio exhibits superior visual quality and better cross-scene consistency. Specifically, VideoStudio surpasses VideoStudio w/o Ref. by 24.3 scene consistency, which essentially verifies the effectiveness of incorporating entity reference images. Moreover, VideoStudio leads to 10.3 and 29.1 improvements in scene consistency over VideoDirectorGPT and Mod-

Table 5: Performance comparisons for multi-scene video generation on Coref-SV.

Approach	CLIPSIM (↑)	Scene Consis. (↑)
ModelScopeT2V [55]	0.3021	37.9
VideoDirectorGPT [26]	-	42.8
VideoStudio w/o Ref.	0.3103	40.9
VideoStudio	**0.3304**	**77.3**

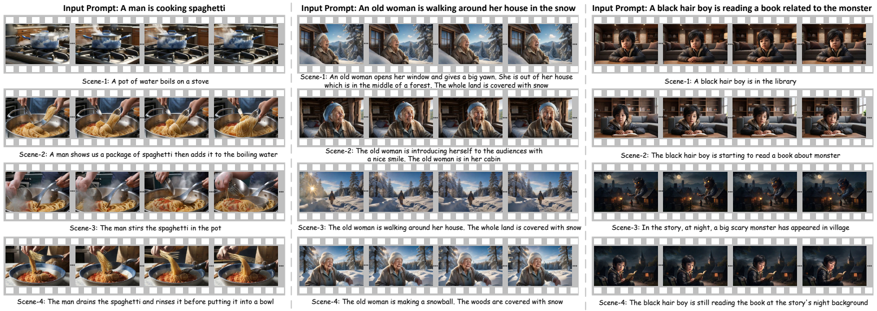

Input Prompt: A man is cooking spaghetti
Scene-1: A pot of water boils on a stove
Scene-2: A man shows us a package of spaghetti then adds it to the boiling water
Scene-3: The man stirs the spaghetti in the pot
Scene-4: The man drains the spaghetti and rinses it before putting it into a bowl
Input Prompt: An old woman is walking around her house in the snow
Scene-1: An old woman opens her window and gives a big yawn. She is out of her house which is in the middle of a forest. The whole land is covered with snow
Scene-2: The old woman is introducing herself to the audiences with a nice smile. The old woman is in her cabin
Scene-3: The old woman is walking around her house. The whole land is covered with snow
Scene-4: The old woman is making a snowball. The woods are covered with snow
Input Prompt: A black hair boy is reading a book related to the monster
Scene-1: A black hair boy is in the library
Scene-2: The black hair boy is starting to read a book about monster
Scene-3: In the story, at night, a big scary monster has appeared in village
Scene-4: The black hair boy is still reading the book at the story's night background

Fig. 7: Examples of generated multi-scene videos by VideoStudio on MSR-VTT. For each video, only the first four scenes are given.

elScopeT2V, respectively. Similar results are also observed on Coref-SV dataset, as summarized in Table 5. Note that as Coref-SV only offers prompts without the corresponding videos, FID and FVD cannot be measured for this case. As shown in the table, VideoStudio again achieves the highest cross-scene consistency of 77.3, making an absolute improvement of 39.4 and 34.5 over ModelScopeT2V and VideoDirectorGPT. Figure 5 showcases an example of generated four-scene videos by different approaches on Coref-SV, manifesting the ability of VideoStudio on generating visually similar entities (e.g., mouse/garden) across scenes. Figure 6 further shows two examples of multi-scene video generation by VideoStudio **using the real images as entity reference images**, which demonstrates the potential of VideoStudio in customizing the generated objects or environments.

4.6 Human Evaluation

Multi-Scene Video Quality. In this section, we conduct a human study to evaluate the entire process of generating multi-scene video from a single prompt. We compare our VideoStudio with four approaches: **ModelScopeT2V w/o LLM** and **VideoStudio w/o Ref. w/o LLM** to generate five scenes by duplicating the input prompt, **ModelScopeT2V w/ LLM** and **VideoStudio w/o Ref.** to utilize LLM to provide video script as described in Sec. 3.1 while generate each scene individually. We invite 12 evaluators and randomly select 100 prompts from MSR-VTT validation set for human evaluation. We show all the evaluators the five videos generated by each approach plus the given prompt and ask them to rank the five videos from 1 to 5 (good to bad) with respect to the three criteria: visual quality (**VQ**), logical coherence (**LC**) and content

Table 6: The user study on three criteria: visual quality (VQ), logical coherence (LC) and content consistency (CC).

Approach	**VQ** (↓)	**LC** (↓)	**CC** (↓)
ModelScopeT2V w/o LLM	4.5	4.7	3.9
ModelScopeT2V w/ LLM	4.5	3.8	4.2
VideoStudio w/o Ref. w/o LLM	2.0	3.0	2.3
VideoStudio w/o Ref.	2.4	2.3	3.4
VideoStudio	**1.6**	**1.2**	**1.2**

Table 7: User preferences on script/videos by using different LLMs in VideoStudio.

	ChatGLM3-6B [8]	GPT-4 [38]	Tie
Video Script	25%	37%	38%
Multi-Scene Video	20%	21%	59%

consistency (**CC**). For each approach, we average the ranking on each criterion of all the generated videos. As indicated by the results in Table 6, the study proves the impact of LLM in generating video script and entity reference images to improve logical coherence and content consistency, respectively. Figure 7 illustrates the examples of the generated multi-scene videos by our VideoStudio.

Different LLMs. To further investigate the effectiveness of different LLMs for multi-scene video generation, we carried out an ablation study on a variant of VideoStudio with GPT-4 [38] in Table 7. Evaluators vote on the preferring video text script by using ChatGLM3-6B and GPT-4, and the corresponding multi-scene videos generated by VideoStudio. "Tie" refers to a close preference. The results indicate that the video script generated by GPT-4 is of higher quality than ChatGLM3-6B. This is not surprising given the significantly larger parameters of GPT-4 ($\sim$1T v.s. 6B). Nevertheless, the voting on multi-scene videos is comparable, showing that the use of an open-source LLM does not affect video quality much. Our exploitation of open-source LLM leads to an elegant view of how responses of light-weight LLM could be improved for video script generation.

5 Conclusions

We have presented a new VideoStudio framework for consistent-content and multi-scene video generation. VideoStudio involves LLM to benefit from the logical knowledge learnt behind and rewrite the input prompt into a multi-scene video script. Then, VideoStudio identifies common entities throughout the script and generates a reference image for each entity, which serves as the link across scenes to ensure the appearance consistency. To produce a multi-scene video, VideoStudio devises two diffusion models of VideoStudio-Img and VideoStudio-Vid. VideoStudio-Img creates a scene-reference image for each scene based on the corresponding event prompt and entity reference images. VideoStudio-Vid converts the scene-reference image into a video clip conditioning on the specific action and camera movement. Extensive evaluations on four video benchmarks demonstrate the superior visual quality and content consistency by VideoStudio over SOTA models.

References

1. Bain, M., Nagrani, A., Varol, G., Zisserman, A.: Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval. In: ICCV (2021)
2. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., Jampani, V., Rombach, R.: Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets. arXiv preprint arXiv:2311.15127 (2023)
3. Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S.W., Fidler, S., Kreis, K.: Align your Latents: High-Resolution Video Synthesis with Latent Diffusion Models. In: CVPR (2023)
4. Carreira, J., Zisserman, A.: Quo Vadis, Action Recognition? A New Model and The Kinetics Dataset. In: CVPR (2017)
5. Chen, H., Xia, M., He, Y., Zhang, Y., Cun, X., Yang, S., Xing, J., Liu, Y., Chen, Q., Wang, X., Weng, C., Shan, Y.: VideoCrafter1: Open Diffusion Models for High-Quality Video Generation. arXiv preprint arXiv:2310.19512 (2023)
6. Chen, Z., Long, F., Qiu, Z., Yao, T., Zhou, W., Luo, J., Mei, T.: Learning Spatial Adaptation and Temporal Coherence in Diffusion Models for Video Super-Resolution. In: CVPR (2024)
7. Dhariwal, P., Nichol, A.: Diffusion Models Beat GANs on Image Synthesis. In: NeurIPS (2021)
8. Du, Z., Qian, Y., Liu, X., Ding, M., Qiu, J., Yang, Z., Tang, J.: GLM: General Language Model Pretraining with Autoregressive Blank Infilling. In: ACL (2022)
9. Esser, P., Chiu, J., Atighehchian, P., Granskog, J., Germanidis, A.: Structure and Content-guided Video Synthesis with Diffusion Models. In: ICCV (2023)
10. Geyer, M., Bar-Tal, O., Bagon, S., Dekel, T.: TokenFlow: Consistent Diffusion Features for Consistent Video Editing. arXiv preprint arXiv:2307.10373 (2023)
11. Guo, Y., Yang, C., Rao, A., Wang, Y., Qiao, Y., Lin, D., Dai, B.: AnimateDiff: Animate Your Personalized Text-to-Image Diffusion Models without Specific Tuning. arXiv preprint arXiv:2307.04725 (2023)
12. He, Y., Yang, T., Zhang, Y., Shan, Y., Chen, Q.: Latent Video Diffusion Models for High-Fidelity Long Video Generation. arXiv preprint arXiv:2211.13221 (2022)
13. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In: NIPS (2017)
14. Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D.P., Poole, B., Norouzi, M., Fleet, D.J., Salimans, T.: Imagen Video: High Definition Video Generation with Diffusion Models. arXiv preprint arXiv:2210.02303 (2022)
15. Ho, J., Jain, A., Abbeel, P.: Denoising Diffusion Probabilistic Models. In: NeurIPS (2020)
16. Ho, J., Salimans, T.: Classifier-Free Diffusion Guidance. arXiv preprint arXiv:2207.12598 (2022)
17. Hong, W., Ding, M., Zheng, W., Liu, X., Tang, J.: CogVideo: Large-Scale Pretraining for Text-to-Video Generation via Transformers. In: ICLR (2023)
18. Hu, Z., Xu, D.: VideoControlNet: A Motion-Guided Video-to-Video Translation Framework by Using Diffusion Model with ControlNet. arXiv preprint arXiv:2307.14073 (2023)
19. Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., Wang, Y., Chen, X., Wang, L., Lin, D., Qiao, Y., Liu, Z.: VBench: Comprehensive Benchmark Suite for Video Generative Models. arXiv preprint arXiv:2311.17982 (2023)

20. Khachatryan, L., Movsisyan, A., Tadevosyan, V., Henschel, R., Wang, Z., Navasardyan, S., Shi, H.: Text2Video-Zero: Text-to-Image Diffusion Models are Zero-Shot Video Generators. In: ICCV (2023)
21. Kim, K.M., Heo, M.O., Choi, S.H., Zhang, B.T.: DeepStory: Video Story QA by Deep Embedded Memory Networks. In: IJCAI (2017)
22. Krishna, R., Hata, K., Ren, F., Fei-Fei, L., Niebles, J.C.: Dense-Captioning Events in Videos. In: ICCV (2017)
23. Li, Y., Yao, T., Pan, Y., Mei, T.: Contextual Transformer Networks for Visual Recognition. IEEE Trans. on PAMI (2022)
24. Li, Y., Gan, Z., Shen, Y., Liu, J., Cheng, Y., Wu, Y., Carin, L., Carlson, D., Gao, J.: StoryGAN: A Sequential Conditional GAN for Story Visualization. In: CVPR (2019)
25. Liang, J., Wu, C., Hu, X., Gan, Z., Wang, J., Wang, L., Liu, Z., Fang, Y., Duan, N.: NUWA-Infinity: Autoregressive over Autoregressive Generation for Infinite Visual Synthesis. In: NeurIPS (2022)
26. Lin, H., Zala, A., Cho, J., Bansal, M.: VideoDirectorGPT: Consistent Multi-Scene Video Generation via LLM-Guided Planning. arXiv preprint arXiv:2309.15091 (2023)
27. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding DINO: Marrying Dino with Grounded Pre-Training for Open-Set Object Detection. arXiv preprint arXiv:2303.05499 (2023)
28. Long, F., Qiu, Z., Pan, Y., Yao, T., Luo, J., Mei, T.: Stand-Alone Inter-Frame Attention in Video Models. In: CVPR (2022)
29. Long, F., Qiu, Z., Pan, Y., Yao, T., Ngo, C.W., Mei, T.: Dynamic Temporal Filtering in Video Models. In: ECCV (2022)
30. Long, F., Yao, T., Qiu, Z., Tian, X., Luo, J., Mei, T.: Gaussian Temporal Awareness Networks for Action Localization. In: CVPR (2019)
31. Long, F., Yao, T., Qiu, Z., Tian, X., Luo, J., Mei, T.: Bi-calibration Networks for Weakly-Supervised Video Representation Learning. IJCV (2023)
32. Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps. In: NeurIPS (2022)
33. Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: DPM-Solver++: Fast Solver for Guided Sampling of Diffusion Probabilistic Models. arXiv preprint arXiv:2211.01095 (2023)
34. Luo, Z., Chen, D., Zhang, Y., Huang, Y., Wang, L., Shen, Y., Zhao, D., Zhou, J., Tan, T.: VideoFusion: Decomposed Diffusion Models for High-Quality Video Generation. In: CVPR (2023)
35. Mou, C., Wang, X., Xie, L., Wu, Y., Zhang, J., Qi, Z., Shan, Y., Qie, X.: T2I-Adapter: Learning Adapters to Dig out More Controllable Ability for Text-to-Image Diffusion Models. arXiv preprint arXiv:2302.08453 (2023)
36. Nichol, A., Dhariwal, P.: Improved Denoising Diffusion Probabilistic Models. In: ICML (2021)
37. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. In: ICML (2022)
38. OpenAI: GPT-4 Technical Report (2023)
39. Ouyang, H., Wang, Q., Xiao, Y., Bai, Q., Zhang, J., Zheng, K., Zhou, X., Chen, Q., Shen, Y.: CoDeF: Content Deformation Fields for Temporally Consistent Video Processing. arXiv preprint arXiv:2308.07926 (2023)
40. Qi, C., Cun, X., Zhang, Y., Lei, C., Wang, X., Shan, Y., Chen, Q.: FateZero: Fusing Attentions for Zero-shot Text-based Video Editing. In: ICCV (2023)

41. Qin, X., Zhang, Z., Huang, C., Dehghan, M., Zaiane, O., Jagersand, M.: U2-Net: Going Deeper with Nested U-Structure for Salient Object Detection. Pattern Recognition (2020)
42. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning Transferable Visual Models from Natural Language Supervision. In: ICML (2021)
43. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical Text-Conditional Image Generation with CLIP Latents. arXiv preprint arXiv:2204.06125 (2022)
44. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-Resolution Image Synthesis with Latent Diffusion Models. In: CVPR (2022)
45. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5B: An Open Large-Scale Dataset for Training Next Generation Image-Text Models. In: NeurIPS (2022)
46. Shin, C., Kim, H., Lee, C.H., gil Lee, S., Yoon, S.: Edit-A-Video: Single Video Editing with Object-Aware Consistency. arXiv preprint arXiv:2303.07945 (2023)
47. Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., Parikh, D., Gupta, S., Taigman, Y.: Make-a-video: Text-to-Video Generation without Text-Video Data. arXiv preprint arXiv:2209.14792 (2022)
48. Sohl-Dickstein, J., Weiss, E.A., Maheswaranathan, N., Ganguli, S.: Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In: ICML (2015)
49. Song, J., Meng, C., Ermon, S.: Denoising Diffusion Implicit Models. In: ICLR (2021)
50. Song, Y., Ermon, S.: Generative Modeling by Estimating Gradients of the Data Distribution. In: NeurIPS (2019)
51. Unterthiner, T., van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., Gelly, S.: FVD: A New Metric for Video Generation. In: ICLR Workshop (2019)
52. Villegas, R., Babaeizadeh, M., Kindermans, P.J., Moraldo, H., Zhang, H., Saffar, M.T., Castro, S., Kunze, J., Erhan, D.: Phenaki: Variable Length Video Generation from Open Domain Textual Description. In: ICLR (2023)
53. Voleti, V., Jolicoeur-Martineau, A., Pal, C.: MCVD-Masked Conditional Video Diffusion for Prediction, Generation, and Interpolation. In: NeurIPS (2022)
54. Wang, F.Y., Chen, W., Song, G., Ye, H.J., Liu, Y., Li, H.: Gen-L-Video: Multi-Text to Long Video Generation via Temporal Co-Denoising. arXiv preprint arXiv:2305.18264 (2023)
55. Wang, J., Yuan, H., Chen, D., Zhang, Y., Wang, X., Zhang, S.: ModelScope Text-to-Video Technical Report. arXiv preprint arXiv:2308.06571 (2023)
56. Wang, W., Yang, H., Tuo, Z., He, H., Zhu, J., Fu, J., Liu, J.: VideoFactory: Swap Attention in Spatiotemporal Diffusions for Text-to-Video Generation. arXiv preprint arXiv:2305.10874 (2023)
57. Wang, X., Yuan, H., Zhang, S., Chen, D., Wang, J., Zhang, Y., Shen, Y., Zhao, D., Zhou, J.: VideoComposer: Compositional Video Synthesis with Motion Controllability. In: NeurIPS (2023)
58. Wu, C., Huang, L., Zhang, Q., Li, B., Ji, L., Yang, F., Sapiro, G., Duan, N.: GODIVA: Generating Open-Domain Videos from Natural Descriptions. arXiv preprint arXiv:2104.14806 (2021)
59. Wu, J.Z., Ge, Y., Wang, X., Lei, S.W., Gu, Y., Shi, Y., Hsu, W., Shan, Y., Qie, X., Shou, M.Z.: Tune-A-Video: One-Shot Tuning of Image Diffusion Models for Text-to-Video Generation. In: ICCV (2023)

60. Xing, J., Xia, M., Zhang, Y., Chen, H., Yu, W., Liu, H., Wang, X., Wong, T.T., Shan, Y.: DynamiCrafter: Animating Open-domain Images with Video Diffusion Priors. arXiv preprint arXiv:2310.12190 (2023)
61. Xu, J., Mei, T., Yao, T., Rui, Y.: MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In: CVPR (2016)
62. Yao, T., Li, Y., Pan, Y., Wang, Y., Zhang, X.P., Mei, T.: Dual Vision Transformer. IEEE Trans. on PAMI (2023)
63. Yao, T., Pan, Y., Li, Y., Ngo, C.W., Mei, T.: Wave-ViT: Unifying Wavelet and Transformers for Visual Representation Learning. In: ECCV (2022)
64. Ye, H., Zhang, J., Liu, S., Han, X., Yang, W.: IP-Adapter: Text Compatible Image Prompt Adapter for Text-to-Image Diffusion Models. arXiv preprint arXiv:2308.06721 (2023)
65. Yin, S., Wu, C., Liang, J., Shi, J., Li, H., Ming, G., Duan, N.: Dragnuwa: Fine-grained Control in Video Generation by Integrating Text, Image, and Trajectory. arXiv preprint arXiv:2308.08089 (2023)
66. Yin, S., Wu, C., Yang, H., Wang, J., Wang, X., Ni, M., Yang, Z., Li, L., Liu, S., Yang, F., Fu, J., Ming, G., Wang, L., Liu, Z., Li, H., Duan, N.: NUWA-XL: Diffusion over Diffusion for eXtremely Long Video Generation. arXiv preprint arXiv:2303.12346 (2023)
67. Zeng, A., Liu, X., Du, Z., Wang, Z., Lai, H., Ding, M., Yang, Z., Xu, Y., Zheng, W., Xia, X., et al.: GLM-130B: An Open Bilingual Pre-Trained Model. arXiv preprint arXiv:2210.02414 (2022)
68. Zhang, L., Rao, A., Agrawala, M.: Adding Conditional Control to Text-to-Image Diffusion Models. In: ICCV (2023)
69. Zhang, S., Wang, J., Zhang, Y., Zhao, K., Yuan, H., Qin, Z., Wang, X., Zhao, D., Zhou, J.: I2VGen-XL: High-Quality Image-to-Video Synthesis via Cascaded Diffusion Models. arXiv preprint arXiv:2311.04145 (2023)
70. Zhang, Z., Long, F., Pan, Y., Qiu, Z., Yao, T., Cao, Y., Mei, T.: TRIP: Temporal Residual Learning with Image Noise Prior for Image-to-Video Diffusion Models. In: CVPR (2024)
71. Zhou, D., Wang, W., Yan, H., Lv, W., Zhu, Y., Feng, J.: Magicvideo: Efficient Video Generation with Latent Diffusion Models. arXiv preprint arXiv:2211.11018 (2022)

VideoStudio: Generating Consistent-Content and Multi-Scene Videos — ECCV 2024 Supplementary Material

Fuchen Long, Zhaofan Qiu, Ting Yao, and Tao Mei

HiDream.ai Inc.
{longfuchen, qiuzhaofan, tiyao, tmei}@hidream.ai

The supplementary material contains: 1) the instructions of LLM; 2) the implementation details of VideoStudio-Img; 3) the implementation details of VideoStudio-Vid; 4) performance contribution of VideoStudio; 5) more video examples generated by VideoStudio; 6) a video demo for VideoStudio.

1 Instructions of LLM

The LLM instructions, output examples and in-context examples for video script and entity description generation are given in Figure 2 and Figure 3, respectively. The multi-round dialogue for entity description generation is shown in Figure 4.

2 Implementation details of VideoStudio-Img

VideoStudio-Img is constructed on Stable Diffusion v2.1 model by incorporating the two additional cross-attention modules. Table 1 details the structures of VideoStudio-Img. We utilize the CLIP ViT-H/14 [42] as the text and visual encoder to extract text features from text prompt, and local image features from foreground/background reference image, respectively. The sequence length L_t of the text features is 77 while the length L_f/L_b of foreground/background image features is set as 256. The cross attention dimension C_t and C_f/C_b are set as the default number in each block of the original diffusion model.

3 Implementation details of VideoStudio-Vid

We build the 3D UNet of VideoStudio-Vid by inserting temporal transformer and temporal convolution layers into 2D UNet of SD-XL. Table 2 details the hyper-parameters and structures of VideoStudio-Vid. We employ the CLIP ViT-H/14 as the visual encoder to extract image features from scene-reference images. To enhance the visual alignment between the scene-reference image and synthesized video, we concatenate the latent code of the scene-reference image with the noisy video latent code along temporal dimension as the input of 3D UNet.

Action condition. In the stage of model training, the VideoMAE [?] fine-tuned on Kinetics-400 [4] is leveraged as the action classifier to measure the action probability (i.e., indicator vector) y_a of input videos. A linear embedding

Table 1: Detailed hyper-parameters and structures of VideoStudio-Img.

Hyper-parameter	Value	Hyper-parameter	Value
Base structure	SD v2.1	Spatial transformer blocks	[1, 1, 1, 1]
Latent shape	$4 \times 64 \times 64$	Image embed sequence	256
Channels	320	Text CLIP	CLIP ViT-H/14
Layers per block	2	Parameterization	ϵ
Channel multiplier	[1, 2, 4, 4]	Diffusion steps	1000
Attention resolutions	[64, 32, 16]	Noise schedule	Scaled Linear
Head channels	[5, 10, 20]	β_1	0.00085
Number of heads	64	β_T	0.0120
CA embed dim	1024	Sampler	DDIM
CA resolutions	[64, 32, 16]	Inference steps	50
Autoencoders	AutoKL	GPU Type	A100-80G
Image CLIP	CLIP ViT-H/14	GPU Number	64
Learning rate	1×10^{-4}	Train steps	20K
Total batch size	512	# of UNet params	915M

Table 2: Detailed hyper-parameters and structures of VideoStudio-Vid.

Hyper-parameter	Value	Hyper-parameter	Value
Base structure	SD-XL	Spatial transformer blocks	[0, 2, 10]
Latent shape	$4 \times 16 \times 40 \times 64$	Temporal transformer blocks	[0, 2, 10]
Channels	320	Temporal SA head number	64
Layers per block	2	Diffusion steps	1000
Channel multiplier	[1, 2, 4]	Noise schedule	Scaled Linear
Attention resolutions	[32, 16]	β_1	0.00085
Head channels	[10, 20]	β_T	0.0120
Number of heads	64	Sampler	DDIM
CA embed dim	1280	Inference steps	70
CA resolutions	[32, 16]	η	1.0
Autoencoders	AutoKL	Guidance scale	12.0
Image CLIP	CLIP ViT-H/14	GPU Type	A100-80G
Parameterization	ϵ	GPU Number	64
Learning rate	3×10^{-6}	Train steps	480K
Total batch size	128	# of 3D-UNet params	4.7B

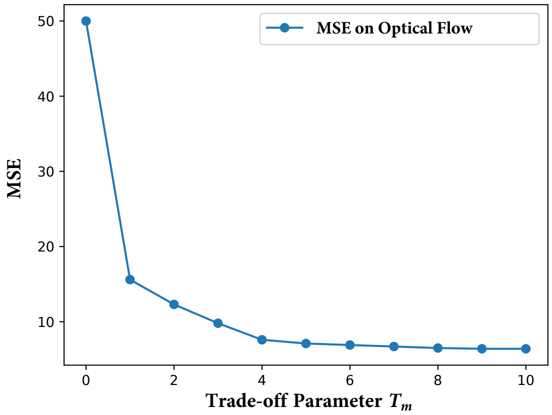


Fig. 1: The impact of trade-off parameter T_m for camera movement.

is then learnt on the probability y_a and further treated as a condition to adjust the video diffusion as demonstrated in the main paper. In inference stage, we use the spaCy library to extract all action phrases from the input text prompt. Next, the text features of the action phrases are obtained by using CLIP model, which are further exploited for cosine similarity computation with action vocabulary of Kinetics-400. For each action phrase, we choose the action category with the max cosine similarity score. If the cosine similarity is lower than 0.2, the action category will be dropped. After collecting all action categories and corresponding cosine similarity, we construct the action indicator vector $y_a \in [0,1]^{400}$ by assigning the normalized cosine similarity into the corresponding category index.

Camera movement. We control the camera movement of each scene video during the inference process of VideoStudio-Vid. Specifically, at inference timestep t, the noisy video $x_t = \alpha_t x_0 + \sigma_t \hat{\epsilon_t}$ is decomposed into the clean video x_0 with an estimated noise $\hat{\epsilon_t} = \epsilon_\theta\left(x_t, t\right)$ with fixed scheduling weights α_t and σ_t. The noisy video x_t is transformed into a video x_{t-1} with reduced noise:

$$x_{t-1} = sampling(x_t, \hat{\epsilon_t}, t), \tag{1}$$

where x_T represents the pure noise ϵ_T. *sampling* is the DDIM [49] update strategy. After the first T_m steps, we execute an adjustment to the noisy video $x_{(T_m-1)}$ to maintain the camera movement indicated by the video script as

$$\begin{aligned} \hat{x_0} &= (x_{(T_m-1)} - \sigma_{(T_m-1)}\hat{\epsilon}_{T_m})/\alpha_{(T_m-1)}, \\ \overline{x_0} &= 0.5 \times \hat{x_0} + 0.5 \times warp(\hat{x_0}, flow), \\ \overline{x_{(T_m-1)}} &= \alpha_{(T_m-1)}\overline{x_0} + \sigma_{(T_m-1)}\epsilon_{\hat{T_m}}, \end{aligned} \tag{2}$$

where $warp(\hat{x_0}, flow)$ is to warp the frames in $\hat{x_0}$ based on the optical flow of required camera movement, and $\overline{x_{(T_m-1)}}$ is the modified noisy video. Such modification ensures that the resultant video clip maintains the same camera movement as we warp the intermediate frames. To analysis the impact of hyperparameter T_m, we conduct the experiments on the MSR-VTT dataset and calculate the mean squared error (MSE) between the optical flow of generated and

Table 3: Performance comparison on ActivityNet Captions dataset.

Approach	Ref	Training Data	Architecture	FID (↓)	FVD (↓)	Scene Consis. (↑)
ModelScopeT2V [55]		LAION + WebVid-10M	SD-2.1	18.1	980	46.0
VideoDirectorGPT [26]		LAION + WebVid-10M + GLIGEN	SD-2.1	16.5	805	64.8
VideoStudio		LAION + WebVid-10M	SD-XL	17.7	789	49.1
		LAION + WebVid-10M + HD-VG	SD-XL	17.3	624	50.8
	✓	LAION + WebVid-10M + HD-VG	SD-XL	**13.2**	**395**	**75.1**

target videos with different T_m, as shown in Figure 1. In general, setting a small T_m for early modification may not effectively control the camera movement, while a late modification may affect the visual quality of the generated videos. As indicated by the figure, T_m=5 provides a good trade-off empirically.

4 Performance Contribution of VideoStudio

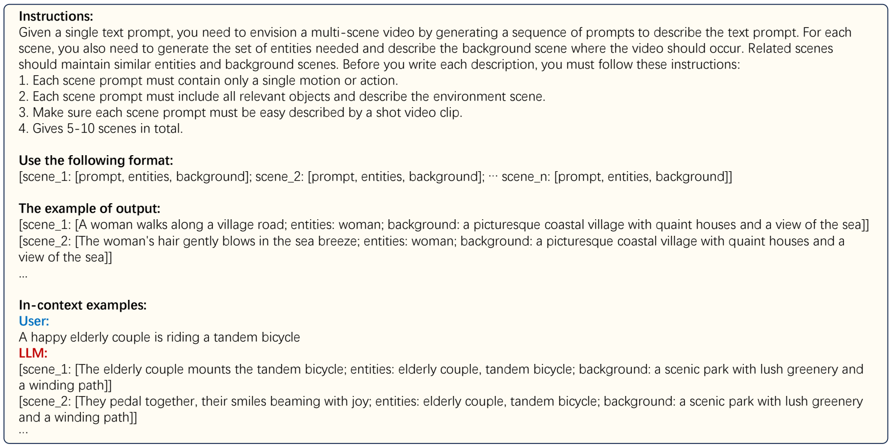
To ablate the performance contribution of VideoStudio more transparent from different perspectives (e.g., with or without reference image, training data and architecture), we report the performances of different VideoStudio variants on ActivityNet Captions dataset in Table 3. The first variant is trained on LAION and WebVid-10M, which is the same as in ModelScopeT2V. The improvement over ModelScopeT2V in video quality (FID & FVD) is due to the deeper Stable Diffusion (SD) backbone and the two-stage (T2I+I2V) framework. The second variant further utilizes HD-VG dataset for model training and leads to slightly better video quality. The full version of VideoStudio takes entity reference images into consideration and improves both video quality and cross-scene consistency.

5 More Video Examples

Here, we present more examples of multi-scene videos generated by VideoStudio on MSR-VTT in Figure 5 and Figure 6 with single foreground reference image and multiple foreground reference images, respectively. For each example, the input prompt, camera movement, foreground/background reference images and generated multi-scene video are given. Figure 7 further showcases three examples of generated multi-scene videos by VideoStudio using the real images as the entity reference images including foreground and background reference images.

6 Video Demo

We have provided a video demo (VideoStudio.mp4) to illustrate the generated videos by VideoStudio in various scenarios. For more details, please refer to the offline project page (VideoStudio.html).



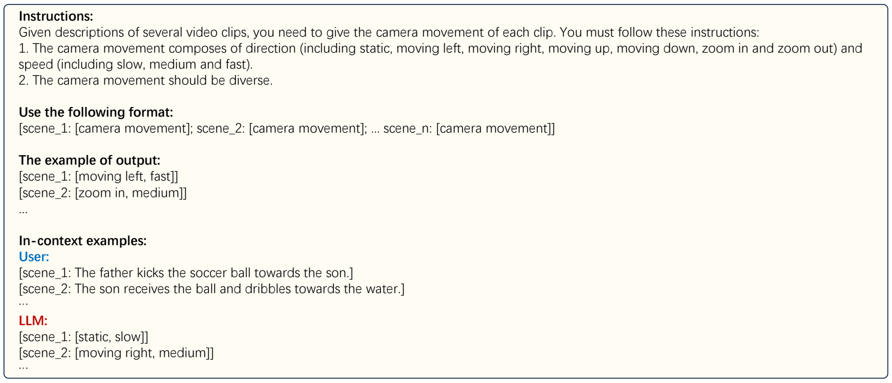
Instructions:
Given a single text prompt, you need to envision a multi-scene video by generating a sequence of prompts to describe the text prompt. For each scene, you also need to generate the set of entities needed and describe the background scene where the video should occur. Related scenes should maintain similar entities and background scenes. Before you write each description, you must follow these instructions:
1. Each scene prompt must contain only a single motion or action.
2. Each scene prompt must include all relevant objects and describe the environment scene.
3. Make sure each scene prompt must be easy described by a shot video clip.
4. Gives 5-10 scenes in total.

Use the following format:
[scene_1: [prompt, entities, background]; scene_2: [prompt, entities, background]; ··· scene_n: [prompt, entities, background]]

The example of output:
[scene_1: [A woman walks along a village road; entities: woman; background: a picturesque coastal village with quaint houses and a view of the sea]]
[scene_2: [The woman's hair gently blows in the sea breeze; entities: woman; background: a picturesque coastal village with quaint houses and a view of the sea]]
...

In-context examples:
User:
A happy elderly couple is riding a tandem bicycle
LLM:
[scene_1: [The elderly couple mounts the tandem bicycle; entities: elderly couple, tandem bicycle; background: a scenic park with lush greenery and a winding path]]
[scene_2: [They pedal together, their smiles beaming with joy; entities: elderly couple, tandem bicycle; background: a scenic park with lush greenery and a winding path]]
...

Fig. 2: The LLM instructions, output examples and in-context examples for generating scene prompts and common entities.



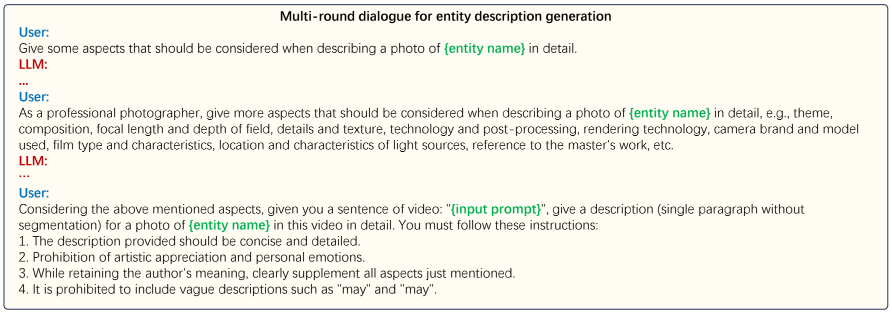
Instructions:
Given descriptions of several video clips, you need to give the camera movement of each clip. You must follow these instructions:
1. The camera movement composes of direction (including static, moving left, moving right, moving up, moving down, zoom in and zoom out) and speed (including slow, medium and fast).
2. The camera movement should be diverse.

Use the following format:
[scene_1: [camera movement]; scene_2: [camera movement]; ... scene_n: [camera movement]]

The example of output:
[scene_1: [moving left, fast]]
[scene_2: [zoom in, medium]]
...

In-context examples:
User:
[scene_1: The father kicks the soccer ball towards the son.]
[scene_2: The son receives the ball and dribbles towards the water.]
...
LLM:
[scene_1: [static, slow]]
[scene_2: [moving right, medium]]
...

Fig. 3: The LLM instructions, output examples and in-context examples for generating camera movements.



Multi-round dialogue for entity description generation
User:
Give some aspects that should be considered when describing a photo of **{entity name}** in detail.
LLM:
...
User:
As a professional photographer, give more aspects that should be considered when describing a photo of **{entity name}** in detail, e.g., theme, composition, focal length and depth of field, details and texture, technology and post-processing, rendering technology, camera brand and model used, film type and characteristics, location and characteristics of light sources, reference to the master's work, etc.
LLM:
...
User:
Considering the above mentioned aspects, given you a sentence of video: "**{input prompt}**", give a description (single paragraph without segmentation) for a photo of **{entity name}** in this video in detail. You must follow these instructions:
1. The description provided should be concise and detailed.
2. Prohibition of artistic appreciation and personal emotions.
3. While retaining the author's meaning, clearly supplement all aspects just mentioned.
4. It is prohibited to include vague descriptions such as "may" and "may".

Fig. 4: The multi-round dialogue of LLM to achieve detailed entity description.

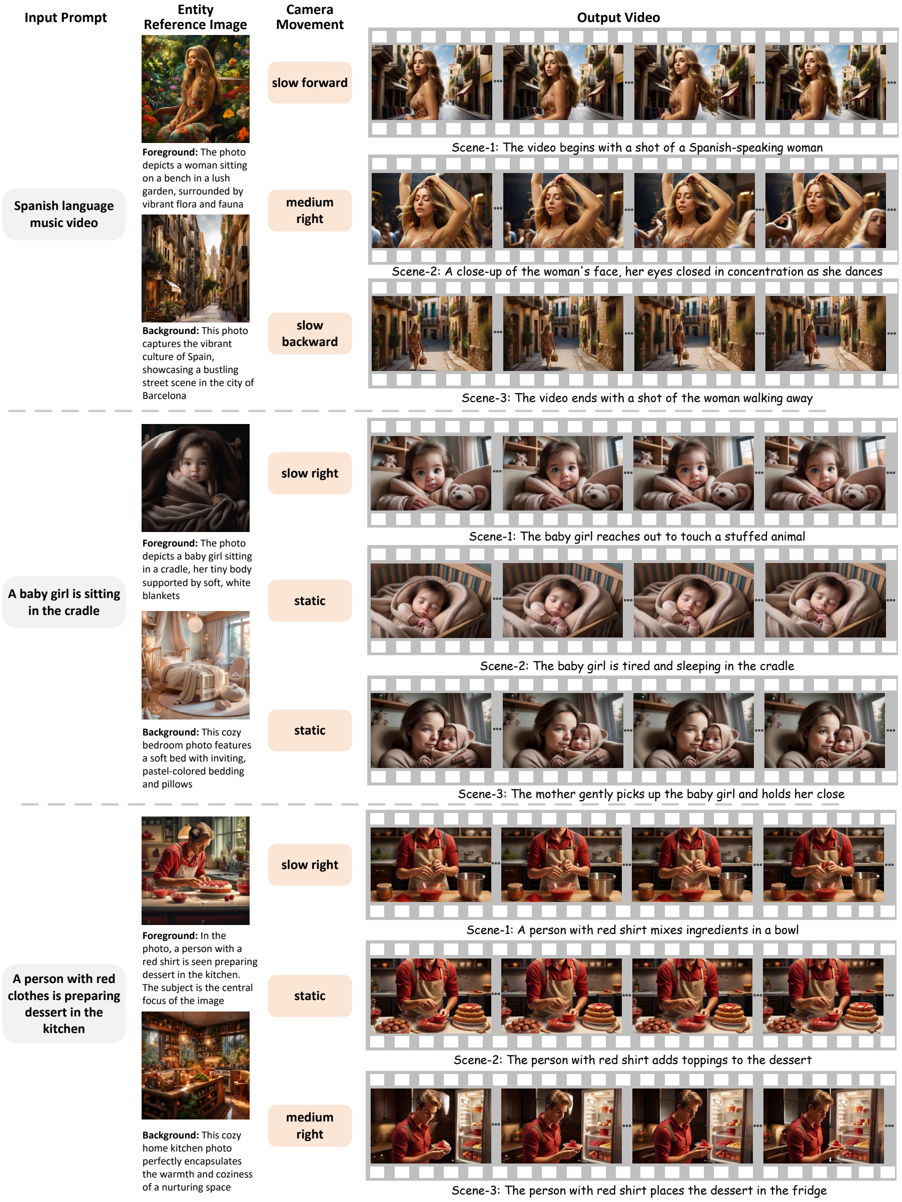

Input Prompt
Entity Reference Image
Camera Movement
Output Video
Spanish language music video
Foreground: The photo depicts a woman sitting on a bench in a lush garden, surrounded by vibrant flora and fauna
Background: This photo captures the vibrant culture of Spain, showcasing a bustling street scene in the city of Barcelona
slow forward
medium right
slow backward
Scene-1: The video begins with a shot of a Spanish-speaking woman
Scene-2: A close-up of the woman's face, her eyes closed in concentration as she dances
Scene-3: The video ends with a shot of the woman walking away
A baby girl is sitting in the cradle
Foreground: The photo depicts a baby girl sitting in a cradle, her tiny body supported by soft, white blankets
Background: This cozy bedroom photo features a soft bed with inviting, pastel-colored bedding and pillows
slow right
static
static
Scene-1: The baby girl reaches out to touch a stuffed animal
Scene-2: The baby girl is tired and sleeping in the cradle
Scene-3: The mother gently picks up the baby girl and holds her close
A person with red clothes is preparing dessert in the kitchen
Foreground: In the photo, a person with a red shirt is seen preparing dessert in the kitchen. The subject is the central focus of the image
Background: This cozy home kitchen photo perfectly encapsulates the warmth and coziness of a nurturing space
slow right
static
medium right
Scene-1: A person with red shirt mixes ingredients in a bowl
Scene-2: The person with red shirt adds toppings to the dessert
Scene-3: The person with red shirt places the dessert in the fridge

Fig. 5: Three examples of generated multi-scene videos by VideoStudio on MSR-VTT with single foreground reference image.

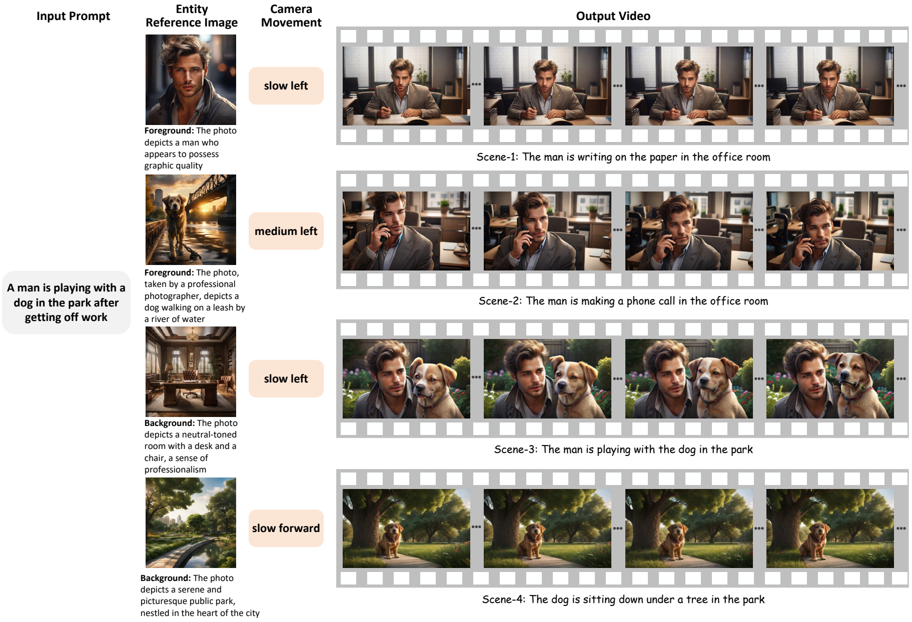

Input Prompt
Entity Reference Image
Camera Movement
Output Video
A man is playing with a dog in the park after getting off work
Foreground: The photo depicts a man who appears to possess graphic quality
slow left
Scene-1: The man is writing on the paper in the office room
Foreground: The photo, taken by a professional photographer, depicts a dog walking on a leash by a river of water
medium left
Scene-2: The man is making a phone call in the office room
Background: The photo depicts a neutral-toned room with a desk and a chair, a sense of professionalism
slow left
Scene-3: The man is playing with the dog in the park
Background: The photo depicts a serene and picturesque public park, nestled in the heart of the city
slow forward
Scene-4: The dog is sitting down under a tree in the park

Fig. 6: One example of generated multi-scene videos by VideoStudio on MSR-VTT with multiple foreground reference images.

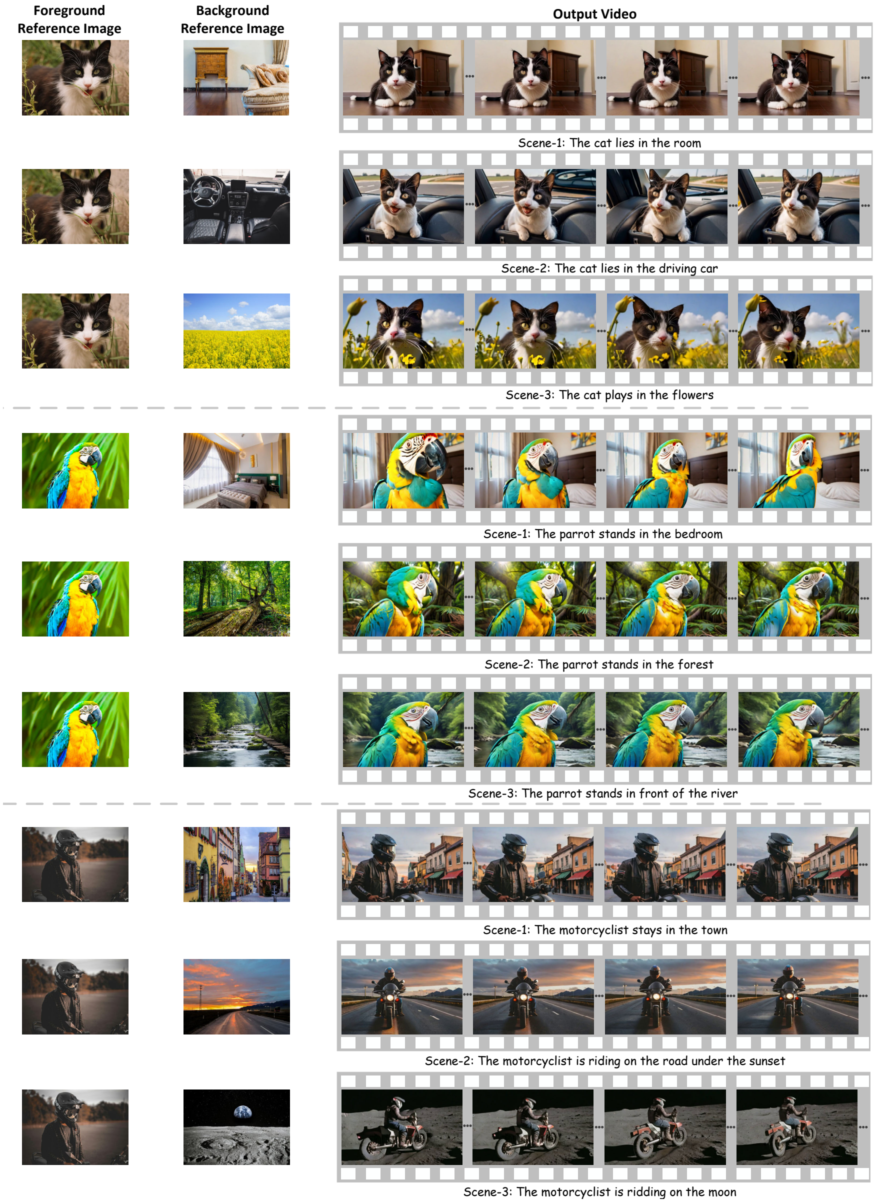

Foreground Reference Image
Background Reference Image
Output Video
Scene-1: The cat lies in the room
Scene-2: The cat lies in the driving car
Scene-3: The cat plays in the flowers
Scene-1: The parrot stands in the bedroom
Scene-2: The parrot stands in the forest
Scene-3: The parrot stands in front of the river
Scene-1: The motorcyclist stays in the town
Scene-2: The motorcyclist is riding on the road under the sunset
Scene-3: The motorcyclist is ridding on the moon

Fig. 7: Three example of generated multi-scene videos by our VideoStudio using the real images as entity reference images.