Modular-Cam: Modular Dynamic Camera-view Video Generation with LLM

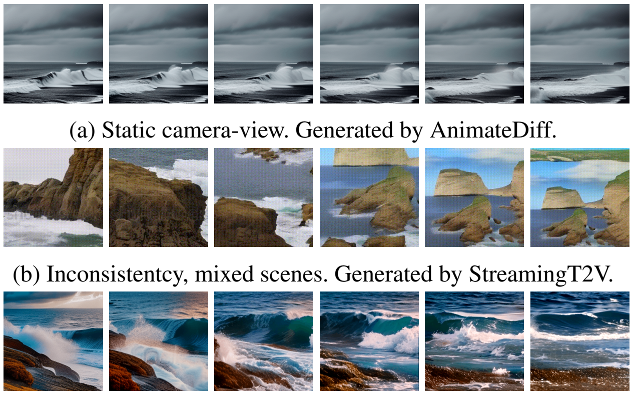
Zirui Pan[1], **Xin Wang**[1,2*], **Yipeng Zhang**[1],
Hong Chen[1], **Kwan Man Cheng**[1], **Yaofei Wu**[3], **Wenwu Zhu**[1,2*]
[1]Department of Computer Science and Technology, Tsinghua University
[2]Beijing National Research Center for Information Science and Technology, Tsinghua University
[3] Beijing University of Technology
{pzr24,zhang-yp22,h-chen20}@mails.tsinghua.edu.cn, kcheng54@wisc.edu,
23027313@emails.bjut.edu.cn,{xin_wang,wwzhu}@tsinghua.edu.cn

Abstract

Text-to-Video generation, which utilizes the provided text prompt to generate high-quality videos, has drawn increasing attention and achieved great success due to the development of diffusion models recently. Existing methods mainly rely on a pre-trained text encoder to capture the semantic information and perform cross attention with the encoded text prompt to guide the generation of video. However, when it comes to complex prompts that contain dynamic scenes and multiple camera-view transformations, these methods can not decompose the overall information into separate scenes, as well as fail to smoothly change scenes based on the corresponding camera-views. To solve these problems, we propose a novel method, i.e., Modular-Cam. Specifically, to better understand a given complex prompt, we utilize a large language model to analyze user instructions and decouple them into multiple scenes together with transition actions. To generate a video containing dynamic scenes that match the given camera-views, we incorporate the widely-used temporal transformer into the diffusion model to ensure continuity within a single scene and propose CamOperator, a modular network based module that well controls the camera movements. Moreover, we propose AdaControlNet, which utilizes ControlNet to ensure consistency across scenes and adaptively adjusts the color tone of the generated video. Extensive qualitative and quantitative experiments prove our proposed Modular-Cam's strong capability of generating multi-scene videos together with its ability to achieve fine-grained control of camera movements. Generated results are available at https://modular-cam.github.io.

Introduction

Via training on large-scale text-image datasets, Text-to-Image (T2I) generation (Rombach et al. 2022; Ramesh et al. 2022; Saharia et al. 2022; Ruiz et al. 2023; Avrahami, Lischinski, and Fried 2022) based on diffusion process has achieved great attention in generating high-quality images with increasing controllability. Due to the significant success of T2I models, many researchers (Ho et al. 2022b,a; Blattmann et al. 2023a; Lu et al. 2023; Chen et al. 2024d,c) have made efforts to take temporal information into considerations for Text-to-Video (T2V) generation. Based on the specific text prompt, T2V models have demonstrated remarkable capability of generating videos that are smooth, photo-realistic, and semantically coherent.



(a) Static camera-view. Generated by AnimateDiff.

(b) Inconsistentcy, mixed scenes. Generated by StreamingT2V.

(c) Generated by this work, which best follows the user instruction.

Figure 1: Generated results based on instruction "*Beginning with a beach scene, the camera gradually draws in closer as waves lap against the reef. Then the camera slowly pans right and a large area of sea is revealed*". In Figure 1a, the video footage is almost static, while in Figure 1b, the scene transitions show abrupt changes, and the scenes are mixed. Figure 1c shows the results of our proposed model.

However, existing works heavily rely on the text encoder to guide the process of generation through capturing semantic information. Due to the limitation of the pre-trained encoder, it is difficult for them to understand the temporal information hidden in the complex text prompts which contain dynamic scenes changes and multiple camera-view transformations. Therefore, these models are not able to disentangle the information of different scenes, failing to sequentially generate these distinct scenes in the prescribed order of camera transformations. Consequently, the generated videos tend to only have a limited amount of motions as well as entangled scenes. Besides, current works mainly focus on generating short videos with merely 16 to 24 frames. Thus, these videos can hardly incorporate all the scenes or characterize the process of the camera transformations between adjacent ones. Although some works (Henschel et al. 2024) adopt an autoregressive method, they fail to achieve fined

[*] Corresponding Author.

control of the camera movements, suffering from color distortion and abrupt transitioning, which is destructive to the realism of the videos.

For instance, consider an instruction shown in Figure 1, which consists of three generated videos. The instruction can be decomposed into two scenes, i.e., i) *Beach, waves lap against the reef* and ii) *large area of sea*, with corresponding transition actions *ZoomIn* and *PanRight*. Figure 1a and Figure 1b demonstrate the lack of fined control of camera movements, as well as inconsistency across multiple scenes with mixed environments and objects, respectively. Figure 1c shows the results of our proposed method, which best follows the instructions. Solving the above problems is challenging since it requires a deep understanding of the instructions and thorough control of the generated content.

To tackle the challenge, in this work we present a novel Modular-Cam framework to address the aforementioned problems. Specifically, to provide a deep understanding of the user instructions, we propose an LLM-Director which utilizes an LLM to analyze the instructions and decompose them into multiple scenes and transition actions. The obtained disentangled information is crucial for the generation of individual scenes and entire videos. Based on T2I diffusion models, we conduct a base video generator by inserting temporal transformer layers, transferring information across frames, and maintaining the continuity of the generated video within a single scene. We further propose a CamOperator module, which is a series of LoRA layers added on the base generator to ensure fine-grained control of the camera movements. For each motion pattern, i.e., *ZoomIn*, *PanLeft*, etc., a corresponding CamOperator module is trained. LLM will select the particular CamOperator module from the operation pool based on the transition action it acquires. Besides, these CamOperators can function as modular components. For complex camera-view transformations, it is not necessary to retrain each of the CamOperators but rather utilize the existing modular operators through their combinations. Benefitting from the modularity, we can easily plug them in at different situations, which greatly enhances the scalability of the model. To improve the consistency across multiple scenes, we adopt an autoregressive method and propose AdaControlNet, which introduces the ending frame of the last scene as the control information for the generation of the current scene and adaptively adjusts the color tone of the videos. Consequently, guided by the last scene, the transition between adjacent scenes will be smooth. We concatenate the video clips for each scene sequentially, deriving the final multi-scene dynamic camera-view video, which completes the end-to-end procedure.

In summary, our contributions can be listed as follows:

- We propose Modular-Cam, which is capable of generating high-quality multi-scene dynamic camera-view videos, ensuring consistency across multiple scenes, and providing a modular method to achieve fine-grained control of the contents and camera movements in the video.
- We propose to use LLM to parse multi-scene involved complicated user instructions, extracting scene descriptions and transition actions, and presenting an end-to-end procedure of generating multi-scene dynamic camera-view videos with modular CamOperators.
- We conduct extensive qualitative and quantitative experiments to verify the strong generating ability of the proposed Modular-Cam method.

Related Work

Text-to-Video Diffusion Models

T2V generation has become popular recently, with large-scale video datasets such as WebVid-10M (Bain et al. 2021) that include about ten million video-text pairs collected from the Internet. Video Diffusion Model (Ho et al. 2022b) is one of the pioneering works in this field which extends a standard text-to-image diffusion model. However, the videos generated have poor resolution. Other works (Singer et al. 2022; Ho et al. 2022a; Blattmann et al. 2023b; Chen et al. 2024b,a; Zhang et al. 2024) improve the quality through video enhancement, specifically by using spatial or temporal upsampling. On this basis, AnimateDiff (Guo et al. 2023b) proposes using a temporal self-attention mechanism to improve frame consistency in a simple and effective way. SparseCtrl (Guo et al. 2023a) further introduces a Control Encoder, adding condition images to the control information. However, many issues remain, such as style-shifting. ModelScopeT2V (Wang et al. 2023b) ensures the consistency of generated videos and the smoothness of object motion within them by incorporating spatial-temporal awareness blocks. Nonetheless, the videos generated by the aforementioned works are still limited in length (mostly about 16 frames), making them more like animated images rather than full-fledged videos.

To generate longer videos, Text2Video-Zero (Khachatryan et al. 2023) still relies on a text-to-image diffusion model and incorporates cross-attention from each frame to the first frame. However, as the number of frames increases, the quality of the generated videos deteriorates, and the motion in the videos remains elementary even static. Gen-L (Wang et al. 2023a) introduces the concept of multi-text, suggesting that a long video may require multiple textual descriptions. FreeNoise (Qiu et al. 2023) adopts a method that requires no additional training, in which it manipulates the initial noise of the diffusion model so that each frame shares a small portion of it and introduces an inter-frame cross-attention mechanism. StreamingT2V (Henschel et al. 2024) employs an autoregressive approach, decomposing long video generation into the generation and stitching of multiple short videos. However, this often results in severe jitter in the visuals, and abrupt transitions may occur between adjacent short videos. Practically, when it comes to multi-scene long video generation, existing works still perform poorly in terms of scene consistency, and fail to achieve fine-grained control of camera-view transformations.

Text-to-Video Generation Guided by LLMs

Given the randomness of the content generated by diffusion models, it is natural to provide some control information to guide the generation process, which is already common in text-to-image diffusion models (Zhang, Rao, and Agrawala

2023). However, for video generation, the control information becomes very complex, potentially requiring descriptions for every scene and even every frame in the video. Recently, with the continuous development of LLMs, several works (Lu et al. 2023; Lian et al. 2023; Long et al. 2024; Lin et al. 2023) have started to explore video generation in complex scenarios using LLMs as a breakthrough point.

LVD (Lian et al. 2023) uses LLMs to generate dynamic scene layouts to assist diffusion models in video generation. This idea actually comes from LayoutGPT (Feng et al. 2024), which uses GPT (Achiam et al. 2023) to generate a series of scene descriptions with multiple bounding boxes based on user instructions. Similar works include FlowZero (Lu et al. 2023), which also utilizes LLMs to parse instructions and generate dynamic scene layouts, introducing a self-refinement process. VideoDirectorGPT (Lin et al. 2023), based on ModelScopeT2V (Wang et al. 2023b), incorporates scene descriptions and object layout information generated by LLMs to improve the controllability of video generation. However, the output multi-scene videos lack smooth transitions, and the generated objects cannot be accurately confined within the bounding boxes.

Other works take a different approach by using LLMs to describe scenes rather than providing layouts. This is for the reason that, in multi-scene video generation, merely providing layouts can become very complex, even for LLMs. VideoDrafter (Long et al. 2024) uses LLMs to parse user instructions containing multiple scenes, generating a text description for each scene and a reference image for each entity. However, the multi-scene videos generated by VideoDrafter are disjointed, with abrupt transitions between adjacent scenes. Free-Bloom (Huang et al. 2024) uses LLMs to generate descriptions for each keyframe in the video, employing joint denoising. Nevertheless, this approach is also challenging in producing long multi-scene videos, as the generation quality tends to degrade when the number of frames increases.

Method

In this section, we will describe our proposed Modular-Cam method. The overall framework is shown in Figure 2. It contains a base video generator which is built upon AnimateDiff (Guo et al. 2023b), a CamOperator, an AdaControlNet, and an LLM-Director. We will introduce a preliminary and give some notations we will use in this paper first and then detail each of the components in the following subsections.

Preliminary

Stable Diffusion Stable Diffusion (Rombach et al. 2022) is a widely adopted model in T2I generation, which is open-sourced and behaves very well, thus we choose it as the base model in Modular-Cam. Stable Diffusion first utilizes a pre-trained encoder $\mathcal{E}(\cdot)$ and a pre-trained decoder $\mathcal{D}(\cdot)$ to encode and decode the image x_0 to and from the latent space, i.e., $z_0 = \mathcal{E}(x_0)$ and $x_0^{'} = \mathcal{D}(z_0)$, respectively, performing the diffusion process in the latent space. In the forward process, the model will gradually add noise to z_0, until we get an approximate Gaussian noise z_T:

$$z_t = \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, 1), \tag{1}$$

where $t = 1, 2, \cdots, T$ represents the steps, and $\bar{\alpha}_t$ stands for the noise strength, ϵ is the added gaussian noise. The process of gradually adding noise is actually a Markov chain process, where we can learn to reverse it by predicting the added noise using a denoising network $\epsilon_\theta(\cdot)$:

$$\mathcal{L} = E_{\mathcal{E}(x_0), y, \epsilon \sim \mathcal{N}(0,1)} \left[||\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y)||_2^2 \right], \tag{2}$$

where y represents the text description corresponding to image x_0, which will be encoded by a CLIP text encoder $\tau_\theta(\cdot)$ (Radford et al. 2021). The text information serves as an input to guide the denoising process. To predict the specific noise, Stable Diffusion utilizes the U-Net (Ronneberger, Fischer, and Brox 2015) which consists of symmetrical encoder and decoder. The encoder is responsible for capturing image information, while the decoder is for merging the control information with the encoded information. Each network block includes stacks of attention layers (Vaswani et al. 2017) and residual mechanism (He et al. 2016). The Base Stable Diffusion model has a large number of parameters, thus we often add LoRA (Hu et al. 2021) layers to finetune it, instead of tuning all the parameters.

Task The main task of this paper is text-to-video generation, i.e., given a text prompt p and the desired length f, generating a series of video frames $x^{1:f}$ that satisfies the requirements of the prompt. Since Stable Diffusion was originally designed for generating images, directly utilizing it to generate videos will perform poorly due to the lack of temporal information. Thus we adopt the widely used approach (Guo et al. 2023b) which inserts temporal transformer layers into the diffusion model to serve as our base video generator.

Suppose we have a noisy latent $z_t^{1:f} \in R^{b \times c \times f \times h \times w}$, where b, c, f, h, w represents the batch size, channel, frame, height and width, respectively. After each pre-trained diffusion layer, we insert a temporal transformer layer to capture temporal information of the latents. Specifically, we first reshape it to $z_t^{(1:f)'} \in R^{f \times (b \times h \times w) \times c}$, and then we perform a self-attention along the frames axis as follows:

$$\begin{aligned} z_t^{out} &= Attention(Query, Key, Value) \\ &= Attention(W^Q z_t^{(1:f)'}, W^K z_t^{(1:f)'}, W^V z_t^{(1:f)'}). \end{aligned} \tag{3}$$

Then the latents are reshaped back and incorporated into the original latents through residual connection. In this way, the temporal transformer layers will adjust the frame vectors by passing information temporally. We finetune the temporal transformer layers while keeping the pre-trained Stable Diffusion network layers fixed on large amounts of video data to learn the continuity across frames.

CamOperator with Modular Network

After conducting the base generator, we further utilize CamOperator, which is a series of tunable LoRA layers adding to

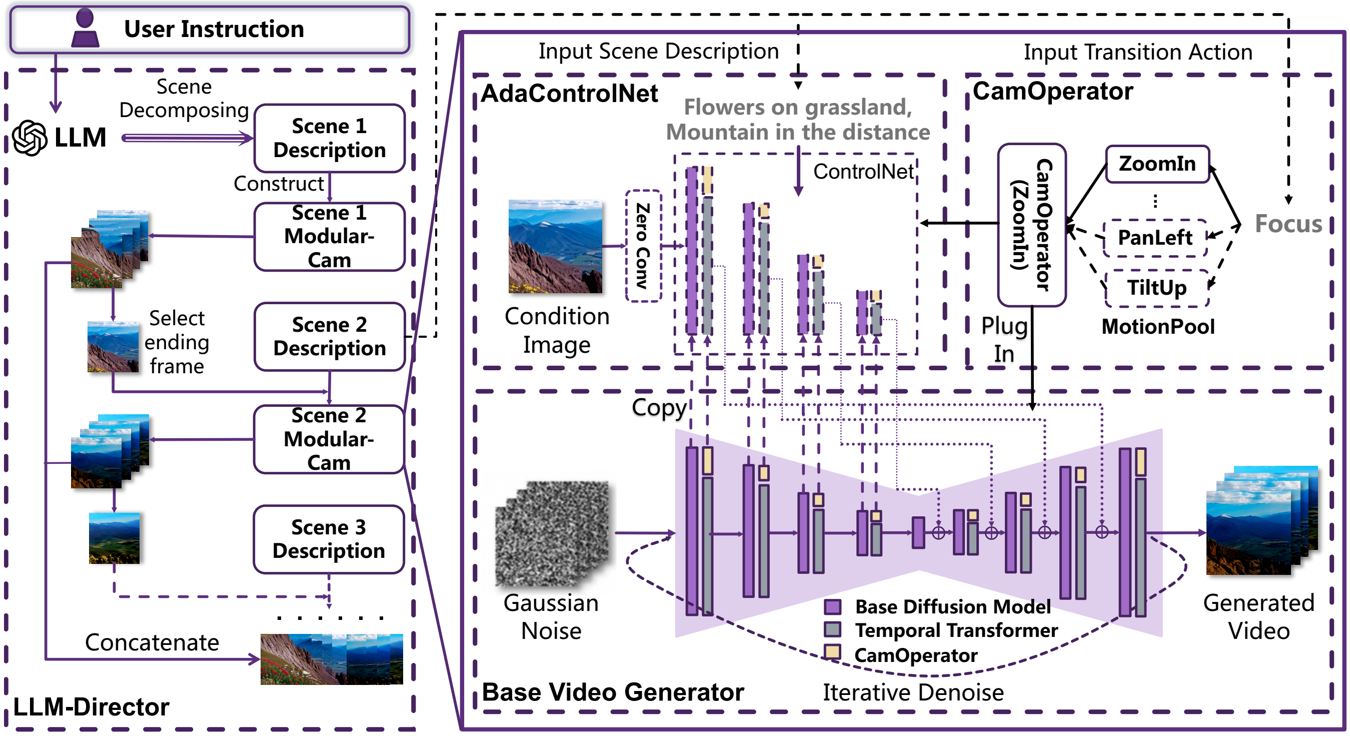


Figure 2: Framework for our proposed Modular-Cam, which contains four modules, i.e., Base Video Generator, CamOperator, AdaControlNet, and LLM-Director. First, the LLM is utilized to parse the user instruction, decomposing it into multiple scenes with descriptions. Then for each scene, a video generator is built, which has been integrated with CamOperator and AdaControlNet. LLM will identify the camera-view transformation in each scene and select from the MotionPool to plug in the appropriate CamOperator Module, which will enable the output video to follow the specific motion pattern, i.e., *ZoomIn*. A condition image, that is the ending frame of the last scene, is inputted into the AdaControlNet, which will guide the generation of the current scene. Finally, the video clips for each scene are concatenated orderly to form the final multi-scene dynamic camera-view video.

the temporal transformer layers, to control the camera movements such as *PanLeft* and *ZoomIn*. We finetune different sets of LoRA parameters for each motion pattern, using generated simulated video data that follow the specific pattern while keeping all the other parameters fixed:

$$\mathcal{W} = \mathcal{W}_{TT} + \Delta\mathcal{W}_{CO} = \mathcal{W}_{TT} + \mathcal{A}_{CO} \times \mathcal{B}_{CO}^{T}, \quad (4)$$

where $\mathcal{W}$ with subscript TT and CO represents parameters for the temporal transformer layer and CamOperator, respectively, and $\mathcal{A}_{CO}$ and $\mathcal{B}_{CO}$ are the low-rank matrix decompositions of $\Delta\mathcal{W}_{CO}$. We simulate the training data with such patterns using data augmentations. For example, for pattern *ZoomIn*, we gradually reduce the video screen size, so that the objects in the video are slowly enlarged, thus creating a zooming-in effect. In this way, we derive CamOperator Module Pool for six basic motion patterns, that is $Motion = \{ZoomIn, ZoomOut, PanLeft, PanRight, TiltUp, TiltDown\}$.

As these modules are trained individually, they can be plugged into the model independently. This modular approach greatly enhances the scalability of the model, for we can train any number of CamOperators at any time. Moreover, due to the low-rank property, these basic modules can be composed to form more complicated motions, such as *PanLeft and ZoomIn*. Thus by decomposing the complicated camera movements into basic ones, we can theoretically simulate any motion pattern in the generated video.

AdaControlNet with Randomized Blending

In the above subsections, we build a complete model for single-scene video generation while the camera movements can be finely controlled. To generate videos with multiple scenes, we propose AdaControlNet and generate videos in an autoregressive manner, which utilizes the ending frame of the last scene as the condition image for current scene generation. However, simply using the controlnet will suffer from the abrupt transitions and the color distortions between adjacent scenes in the video. To solve the problem, we perform an adaptive pixel normalization to the controlnet which adjusts the color tone between different scenes. To further enhance the consistency between the starting frame of the video clip for the current scene and the ending frame of the last scene, we utilize a randomized blending technique inspired by (Avrahami, Lischinski, and Fried 2022). Thus, the generated multi-scene video can satisfy the semantic requirements while having excellent consistency.

First, we duplicate the structure and parameters of the encoder in the U-Net as our AdaControlNet. Similarly, we apply the temporal attention mechanism into the AdaControlNet so that the condition image will not only affect the starting frame but influence the rest frames as well. And we replace the input of the AdaControlNet from the concatenation of the encoded condition image and noisy latent $z_t^{1:f}$ to the encoded condition image alone, to remove the harming effects of $z_t^{1:f}$ on the AdaControlNet. We derive our training data by selecting the first frame of the video data as the condition image and finetune the AdaControlNet while keeping all the other parameters fixed.

We find that simply using the Controlnet can ensure the consistency of objects and layouts across multiple scenes, but the color tone of the generated video may drift from the condition image, i.e., become darker or lighter. Visually, such a difference is easily recognized by the naked eye, which reduces the authenticity of the generated video. Therefore, we perform an adaptive pixel normalization which adjusts the mean and variance of the three color channels (RGB) of the generated video on the pixel-level to make it consistent with the condition image:

$$frame^{ch} = \frac{frame^{ch} - frame^{ch}_{mean}}{frame^{ch}_{std}} \cdot cond^{ch}_{std} + cond^{ch}_{mean}, \tag{5}$$

where $frame$ and $cond$ represent a frame image or a condition image. Superscript ch stands for color channel $\in \{R, G, B\}$. Additionally, we propose to use randomized blending to further unify the color tone of the generated video:

$$z_t^1 = \begin{cases} z_t^{cond}, & if \ \texttt{random}\,(0,1) < \lambda \\ z_t^1, & otherwise, \end{cases} \tag{6}$$

where z_t^1 and z_t^{cond} represent the tth noisy latent for the first frame of the current scene and the condition image. $\texttt{random}(0,1)$ generates a random number uniformly distributed in $(0,1)$, and λ is a hyper-parameter controlling the intensity of randomized blending. The essence is that z_t^1 will receive z_t^{cond} with the probability λ. We can derive that larger λ will introduce more blending, thus the first frame will be more like the condition image, improving consistency. However, frequent blending will reduce the continuity between the first frame and the rest. On the other hand, a small λ will encourage free generation, which improves overall continuity. In practical inference, we set λ to 0.5.

By combining the two techniques of adaptive pixel normalization and random blending during inference, the transitions between scenes become smooth, and the multi-scene video maintains the consistency of color tone and content.

LLM Director with Modularized Motion Selection

Integrating AdaControlNet, we can now generate a multi-scene dynamic camera-view video. However, for a complicated multi-scene involved user instruction, the scene description or transition action may not be given directly, and the video generation model cannot automatically extract all the information. Therefore, we utilize LLM to parse user instruction, decomposing it into different scenes, and extract the transition actions between adjacent ones. LLM is like a director, guiding the video generation model in producing multi-scene videos.

Take the user instruction “Starting with a long shot of a field and blue sky, and gradually focusing on a house in the distance. Then the camera moves to the left, and large fields appear, then the house moves out of view” as an example, which has two scenes. We design the prompt:

- ”*Extract the scenes that appear in the given text in order, and identify the transition actions between adjacent scenes. The scene description should contain rich information. You should pick the transition action from [Zoom In, Zoom Out, Pan Left, Pan Right, Tilt Up, Tilt Down]*”.

LLM will analyze the user instruction, decomposing it into the two scenes and output to the specific format for the video generation model to receive:

$$\begin{aligned} &[Scene1 : \text{“field and blue sky, house in the distance”},\\ &\ Action : \text{ZoomIn}] \\ &[Scene2 : \text{“large fields”}, Action : \text{PanLeft}]. \end{aligned} \tag{7}$$

For *Scene1*, our video generator will select and integrate *ZoomIn* CamOperator Module to generate a single scene video. And for *Scene2*, our video generator will select the last frame of *Scene1* as a condition image and integrate *PanLeft* CamOperator Module for the generation. The final video is derived by concatenating video clips for *Scene1* and *Scene2*.

Note that although LLM is not directly involved in specific video generation, it still plays an indispensable role, just like what an excellent director can bring to a film. If we input the user instruction directly to the video generation model without LLM parsing, the scenes in the generated video will be mixed.

Thus, we get Modular-Cam, capable of generating dynamic camera-view transformations and multi-scene long videos based on complex user instructions.

Experiment

In this section, we first detail on the specific setting of the training and testing of our proposed Modular-Cam, and conduct extensive quantitative and qualitative experiments to demonstrate the strong generating ability of our model. We further conduct some ablation studies to verify the effectiveness of each module.

Experiment Setup

We use the large-scale public video dataset WebVid-10M (Bain et al. 2021) as our training set to train the newly inserted temporal transformer layers. For CamOperator, we simulate and generate about 50 videos with specific motion patterns to finetune the LoRA layers. For AdaControlNet, we select 100,000 videos from WebVid-10M as the training set, with the starting frame as the condition image. The whole training procedure can be found in the Appendix.

Model	MS(↑)	DD(↑)	IQ(↑)	CLIP(↑)	UR(↓)
AnimateDiff	0.983	0.329	**0.622**	0.171	3.8
FreeNoise	0.986	0.302	0.618	0.171	3.7
SparseCtrl	0.952	0.677	0.524	0.208	3.5
StreamingT2V	0.974	0.907	0.350	0.224	2.3
Modular-Cam	**0.988**	**0.994**	0.546	**0.232**	**1.7**

Table 1: Quantitative comparison between Modular-Cam and other baselines, where MS, DD and IQ stands for *Motion Smoothness*, *Dynamic Degree* and *Imaging Quality*, respectively, and UR represents *User Rank*, a user evaluation metric. The top and second top performances have been bolded or underlined, respectively. ↑ represents that the higher the metric, the better, while ↓ represents the opposite.

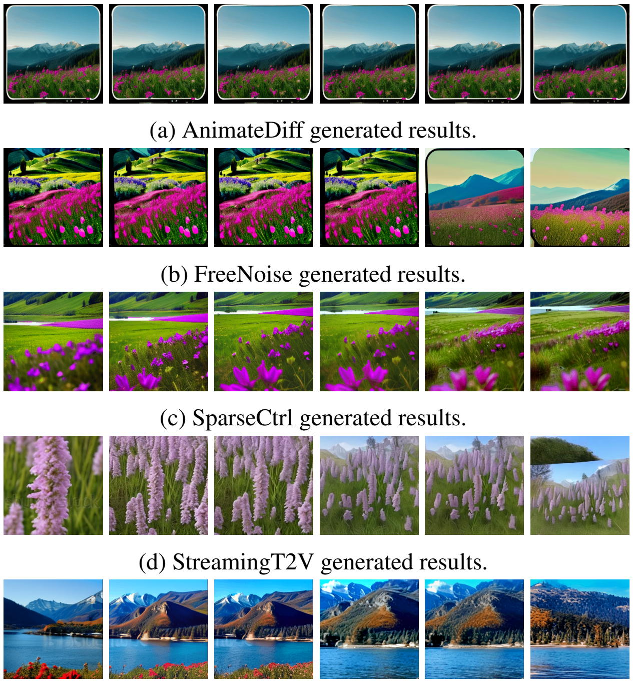
In the inference stage, since our work mainly focuses on multi-scene dynamic camera-view video generation, current instruction sets, which mostly contain simple single-scene instructions, cannot satisfy our requirements. Therefore, in quantitative comparison, we adopt a self-generated dataset, which contains 1000 multi-scene involved instructions. We use ChatGPT3.5-turbo (Achiam et al. 2023) to auto-generate the dataset, ensuring that each instruction has at least two scenes with guidance on camera-view transformations.

We compare Modular-Cam with the baselines in terms of five metrics, i.e., *Motion Smoothness*(MS), *Dynamic Degree*(DD), *Imaging Quality*(IQ), CLIP Metric and *User Rank*(UR). The detailed information of the employed metrics can be found in the appendix.

Main Results

We conduct quantitative and qualitative comparisons and demonstrate the results in Table 1 and Figure 3. In quantitative comparison, we can observe that Modular-Cam outperforms the other baselines in most of the metrics, no matter computed or manually evaluated, and is only slightly lower than AnimateDiff and FreeNoise under *IQ*, which may be attributed to the trade-off between content richness and quality, since Modular-Cam generates a complicated video with multi-scenes and camera-view transformations, and is relatively harder to maintain the same-level quality. In terms of *Dynamic Degree*, Modular-Cam is much higher than the other baselines, validating its superior dynamic camera-view generating ability. StreamingT2V also focuses on multi-scene video generation and performs similarly to Modular-Cam in terms of *CLIP Metric* and *MS*. However, it falls behind in *UR* and *DD*, and drops drastically on *IQ*.

In qualitative comparison, we evaluate Modular-Cam and other baselines based on the same user instruction, that is *"Starting with a close up shot of the flowers in the meadow, the camera slowly moves to the right to focus on the mountain peaks in the distance and gradually draws in closer. The camera then continues to move to the right as the mountain and the lake mirror each other"*. The results are shown in Figure 3. The instruction can be decomposed into three scenes, i.e., *close up shot of the flowers in the meadow*, *gradually focus on the mountain in the distance* and *mountain and lake mirror each other*, with transition actions *PanRight*, *ZoomIn* and *PanRight*, respectively. We can observe that the videos generated by AnimateDiff, FreeNoise and SparseCtrl are almost static in motion, especially in results generated by FreeNoise, the frames change completely in the later stage, potentially affected by the *ZoomIn* instruction, while StreamingT2V displays abrupt transitions, with mixed scenes in the end, i.e., mountain and flower. On the other hand, the video outputted by Modular-Cam best follows the user instructions, generating all the objects correctly and performing the right camera-view transformations. Additional results can be found in the Appendix.


(a) AnimateDiff generated results.

(b) FreeNoise generated results.

(c) SparseCtrl generated results.

(d) StreamingT2V generated results.

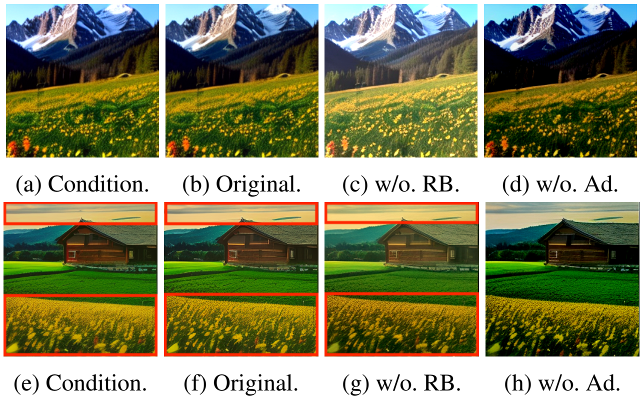
(e) Modular-Cam generated results.

Figure 3: Qualitative comparison between Modular-Cam and other baselines. We select several intermediate frames of the whole video for the convenience of presentation.

Ablation Studies

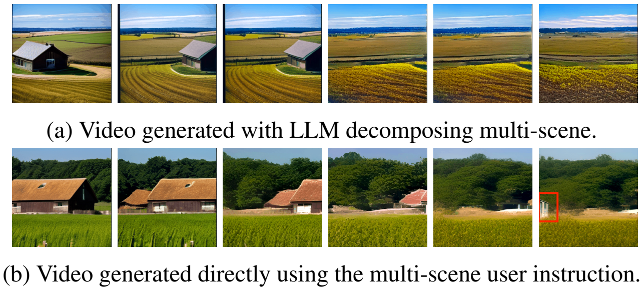
In this subsection, we validate the effectiveness of the proposed modules through a series of ablation studies.

AdaControlNet Due to the misalignment of data distributions in training and inference, color tone shifting usually occurs in the generated results of diffusion models (Song and Ermon 2020). Therefore, we propose to use *Adaptive pixel normalization* and *Randomized blending* to mitigate the shifting problem. From the results in Figure 4, we can observe that without *Randomized blending*, the frames become slightly whiter or darker, as has been marked with the red frame, while without *Adaptive pixel normalization*, the frames overall obviously turn darker. The change of color tone, even the smallest, may possibly be recognized by the


(a) Condition. (b) Original. (c) w/o. RB. (d) w/o. Ad.

(e) Condition. (f) Original. (g) w/o. RB. (h) w/o. Ad.

Figure 4: Ablation study on adjusting the color tone, where *Original* represents the generated results of Modular-Cam, and *RB* and *Ad* stands for *Randomized blending* and *Adaptive pixel normalization*, respectively. We remove the two techniques and display the first frame of each generated video compared with the condition image, where areas with color tone shifting are marked with red frames.

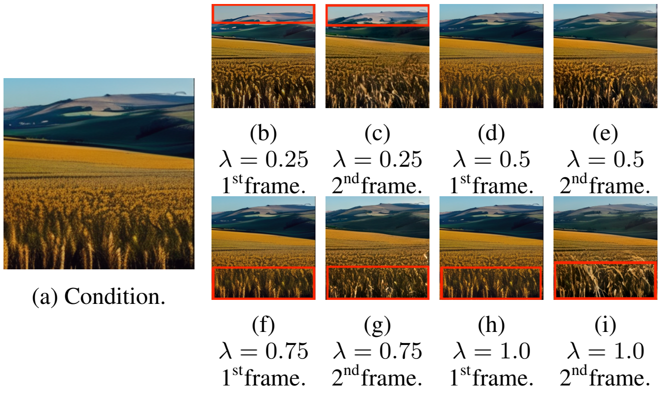

(a) Video generated with LLM decomposing multi-scene.

(b) Video generated directly using the multi-scene user instruction.

Figure 5: Ablation study on LLM decomposing user instruction. In Modular-Cam, the instruction is first parsed and decomposed by LLM then be fed to the video generator, while in Figure 5b, we display the generated result of video generator directly utilizing the multi-scene involved instruction.

naked eye, thus destructing the general realism. Utilizing the two techniques, the generated results of Modular-Cam showcase the most aligned color tone, which enhances the authenticity of the generated video.

LLM-Director To illustrate the important role of LLM decomposing multi-scene involved instructions, we design a simple prompt "*Beginning with a scene of fields and houses, the camera gradually moves to the left, the houses move out of view, and large fields appear*", which consists of only two scenes, with transition action *PanLeft*. We parse it with LLM and obtain the decomposed two scenes as *fields and house* and *large fields*, in which we can find that the LLM has understood the instruction to remove the object *house* out of the second scene, avoiding confusing the video generator. We compare it with the generated video of directly utilizing the undecomposed instruction, i.e., the description for each scene is the original multi-scene user instruction. The results are in Figure 5. We can observe that in Figure 5b, mixed scenes occur in scene 2, where objects similar to *house* appear again in the scene, as has been marked with the red frame, which is contrary to the *move out of view* instruction, while in Figure 5a only large fields remain, showcasing the strong comprehension capability brought by the LLM.


(a) Condition.

(b) $\lambda = 0.25$ 1$^{\text{st}}$frame. (c) $\lambda = 0.25$ 2$^{\text{nd}}$frame. (d) $\lambda = 0.5$ 1$^{\text{st}}$frame. (e) $\lambda = 0.5$ 2$^{\text{nd}}$frame.

(f) $\lambda = 0.75$ 1$^{\text{st}}$frame. (g) $\lambda = 0.75$ 2$^{\text{nd}}$frame. (h) $\lambda = 1.0$ 1$^{\text{st}}$frame. (i) $\lambda = 1.0$ 2$^{\text{nd}}$frame.

Figure 6: Sensitivity analysis on the λ introduced in randomized blending. Here we only display the 1$^{\text{st}}$ and 2$^{\text{nd}}$ frame of the generated results for specific value of λ. We mark the areas where inconsistent color tone or transition gap occurs with red frame.

Parameter Sensitivity

In Equation 6, we introduce a hyper-parameter λ to control the intensity of randomized blending. We determine the optimal value of λ through sensitivity analysis. Since randomized blending only directly impact on the 1$^{\text{st}}$ frame, we can judge the continuity and consistency of the video by comparing the 1$^{\text{st}}$ frame with the condition image and the 2$^{\text{nd}}$ frame, which represents the rest of the video, respectively. From Figure 6, we can observe that small λ reduces consistency with the condition image in terms of color tone, while large λ results in transition gap between the 1st frame and the rest of the video, where misaligned shapes and positions of objects occur, which confirms our presumption. Experimentally, we find that $\lambda = 0.5$ achieves the best balance between the continuity within a single scene and the consistency across multiple scenes.

Conclusion

In this work, we present a novel method called Modular-Cam, which is able to generate multi-scene dynamic camera-view video, overcoming the limitations of existing works, which either output videos that are almost static, without much motion dynamics, or produce severe gaps between adjacent scenes. We propose three modules to address these problems, namely CamOperator, AdaControlNet, and LLM-Director, to enhance the consistency across multiple scenes and provide fine-grained control of camera movements, where we utilize modular network to learn each motion pattern and take advantage of LLM's understanding capacity to guide the video generation. Extensive experiments verify the strong generating ability of Modular-Cam and the effectiveness of each proposed module.

Acknowledgement

This work was supported by the National Key Research and Development Program of China No.2023YFF1205001, National Natural Science Foundation of China No. 62222209, Beijing National Research Center for Information Science and Technology under Grant No. BNR2023TD03006, BNR2023RC01003, and Beijing Key Lab of Networked Multimedia.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Avrahami, O.; Lischinski, D.; and Fried, O. 2022. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 18208–18218.

Bain, M.; Nagrani, A.; Varol, G.; and Zisserman, A. 2021. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1728–1738.

Blattmann, A.; Dockhorn, T.; Kulal, S.; Mendelevitch, D.; Kilian, M.; Lorenz, D.; Levi, Y.; English, Z.; Voleti, V.; Letts, A.; et al. 2023a. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*.

Blattmann, A.; Rombach, R.; Ling, H.; Dockhorn, T.; Kim, S. W.; Fidler, S.; and Kreis, K. 2023b. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22563–22575.

Chen, H.; Wang, X.; Zeng, G.; Zhang, Y.; Zhou, Y.; Han, F.; Wu, Y.; and Zhu, W. 2024a. Videodreamer: Customized Multi-subject Text-to-video Generation with Disen-mix Finetuning on Language-Video Foundation Models. *IEEE Transactions on Multimedia*.

Chen, H.; Wang, X.; Zhang, Y.; Zhou, Y.; Zhang, Z.; Tang, S.; and Zhu, W. 2024b. Disenstudio: Customized multi-subject text-to-video generation with disentangled spatial control. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 3637–3646.

Chen, H.; Zhang, Y.; Wang, X.; Duan, X.; Zhou, Y.; and Zhu, W. 2024c. DisenDreamer: Subject-Driven Text-to-Image Generation with Sample-aware Disentangled Tuning. *IEEE Transactions on Circuits and Systems for Video Technology*.

Chen, H.; Zhang, Y.; Wu, S.; Wang, X.; Duan, X.; Zhou, Y.; and Zhu, W. 2024d. Disenbooth: Identity-preserving disentangled tuning for subject-driven text-to-image generation. *International conference on learning representations*.

Feng, W.; Zhu, W.; Fu, T.-j.; Jampani, V.; Akula, A.; He, X.; Basu, S.; Wang, X. E.; and Wang, W. Y. 2024. Layoutgpt: Compositional visual planning and generation with large language models. *Advances in Neural Information Processing Systems*, 36.

Guo, Y.; Yang, C.; Rao, A.; Agrawala, M.; Lin, D.; and Dai, B. 2023a. Sparsectrl: Adding sparse controls to text-to-video diffusion models. *arXiv preprint arXiv:2311.16933*.

Guo, Y.; Yang, C.; Rao, A.; Wang, Y.; Qiao, Y.; Lin, D.; and Dai, B. 2023b. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Henschel, R.; Khachatryan, L.; Hayrapetyan, D.; Poghosyan, H.; Tadevosyan, V.; Wang, Z.; Navasardyan, S.; and Shi, H. 2024. Streamingt2v: Consistent, dynamic, and extendable long video generation from text. *arXiv preprint arXiv:2403.14773*.

Ho, J.; Chan, W.; Saharia, C.; Whang, J.; Gao, R.; Gritsenko, A.; Kingma, D. P.; Poole, B.; Norouzi, M.; Fleet, D. J.; et al. 2022a. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*.

Ho, J.; Salimans, T.; Gritsenko, A.; Chan, W.; Norouzi, M.; and Fleet, D. J. 2022b. Video diffusion models. *Advances in Neural Information Processing Systems*, 35: 8633–8646.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; and Chen, W. 2021. LoRA: Low-Rank Adaptation of Large Language Models. *CoRR*, abs/2106.09685.

Huang, H.; Feng, Y.; Shi, C.; Xu, L.; Yu, J.; and Yang, S. 2024. Free-bloom: Zero-shot text-to-video generator with llm director and ldm animator. *Advances in Neural Information Processing Systems*, 36.

Khachatryan, L.; Movsisyan, A.; Tadevosyan, V.; Henschel, R.; Wang, Z.; Navasardyan, S.; and Shi, H. 2023. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15954–15964.

Lian, L.; Shi, B.; Yala, A.; Darrell, T.; and Li, B. 2023. Llm-grounded video diffusion models. *arXiv preprint arXiv:2309.17444*.

Lin, H.; Zala, A.; Cho, J.; and Bansal, M. 2023. Videodirectorgpt: Consistent multi-scene video generation via llm-guided planning. *arXiv preprint arXiv:2309.15091*.

Long, F.; Qiu, Z.; Yao, T.; and Mei, T. 2024. Videodrafter: Content-consistent multi-scene video generation with llm. *arXiv preprint arXiv:2401.01256*.

Lu, Y.; Zhu, L.; Fan, H.; and Yang, Y. 2023. Flowzero: Zero-shot text-to-video synthesis with llm-driven dynamic scene syntax. *arXiv preprint arXiv:2311.15813*.

Qiu, H.; Xia, M.; Zhang, Y.; He, Y.; Wang, X.; Shan, Y.; and Liu, Z. 2023. Freenoise: Tuning-free longer video diffusion via noise rescheduling. *arXiv preprint arXiv:2310.15169*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2): 3.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, 234–241. Springer.

Ruiz, N.; Li, Y.; Jampani, V.; Pritch, Y.; Rubinstein, M.; and Aberman, K. 2023. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 22500–22510.

Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E. L.; Ghasemipour, K.; Gontijo Lopes, R.; Karagol Ayan, B.; Salimans, T.; et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35: 36479–36494.

Singer, U.; Polyak, A.; Hayes, T.; Yin, X.; An, J.; Zhang, S.; Hu, Q.; Yang, H.; Ashual, O.; Gafni, O.; et al. 2022. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*.

Song, Y.; and Ermon, S. 2020. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33: 12438–12448.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, F.-Y.; Chen, W.; Song, G.; Ye, H.-J.; Liu, Y.; and Li, H. 2023a. Gen-l-video: Multi-text to long video generation via temporal co-denoising. *arXiv preprint arXiv:2305.18264*.

Wang, J.; Yuan, H.; Chen, D.; Zhang, Y.; Wang, X.; and Zhang, S. 2023b. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*.

Zhang, L.; Rao, A.; and Agrawala, M. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3836–3847.

Zhang, Y.; Wang, X.; Chen, H.; Qin, C.; Hao, Y.; Mei, H.; and Zhu, W. 2024. ScenarioDiff: Text-to-video Generation with Dynamic Transformations of Scene Conditions. *International Journal of Computer Vision (IJCV) 2024*.