International Journal of Computer Vision
https://doi.org/10.1007/s11263-025-02413-7

ScenarioDiff: Text-to-video Generation with Dynamic Transformations of Scene Conditions

Yipeng Zhang[1] · Xin Wang[1,2] · Hong Chen[1] · Chenyang Qin[1] · Yibo Hao[1] · Hong Mei[3] · Wenwu Zhu[1,2]

Received: 29 July 2024 / Accepted: 10 October 2024

Abstract

With the development of diffusion models, text-to-video generation has recently received significant attention and achieved remarkable success. However, existing text-to-video approaches suffer from the following weaknesses: i) they fail to control the trajectory of the subject as well as the process of scene transformations; ii) they can only generate videos with limited frames, failing to capture the whole transformation process. To address these issues, we propose the model named ScenarioDiff, which is able to generate longer videos with scene transformations. Specifically, we employ a spatial layout fuser to control the positions of subjects and the scenes of each frame. To effectively present the process of scene transformation, we introduce mixed frequency controlnet, which utilizes several frames of the generated videos to extend them to long videos chunk by chunk in an auto-regressive manner. Additionally, to ensure consistency between different video chunks, we propose a cross-chunk scheduling mechanism during inference. Experimental results demonstrate the effectiveness of our approach in generating videos with dynamic scene transformations. Our project page is available at https://scenariodiff2024.github.io/.

Keywords Text-to-video generation · Diffusion · Controllable generation · AIGC

1 Introduction

With the advancement of the diffusion methods [15, 22, 38], text-to-image generation has achieved significant success in recent years [6, 33, 35]. Boosted by the success of image generation, researchers have made numerous efforts to incorporate temporal information into the diffusion models for text-to-video generation. The text-to-video diffusion models are trained on large-scale multi-modal datasets [3, 12, 16, 27, 37, 45], which have demonstrated remarkable ability to generate videos that are semantically coherent, photo-realistic, and consistent with the given text prompts.

Nevertheless, existing text-to-video approaches suffer from two issues, i.e., failing to control the trajectory of the subject together with the process of scene transformations, and failing to capture the whole transformation process due to the generation of videos with limited frames. More concretely, relying solely on textual information poses challenges in controlling specific details, such as the motion trajectory of the subject and the transitions between scenes or backgrounds in text-to-video generation, and in Fig. 1, we provide several examples to illustrate these problems. Take the text prompt "*a dog is running from grass to the wheat field*" as an example. An expected result requires that most

Communicated by Long Yang.

✉ Xin Wang
xin_wang@tsinghua.edu.cn

✉ Wenwu Zhu
wwzhu@tsinghua.edu.cn

Yipeng Zhang
zhang-yp22@mails.tsinghua.edu.cn

Hong Chen
h-chen20@mails.tsinghua.edu.cn

Chenyang Qin
qcy22@mails.tsinghua.edu.cn

Yibo Hao
haoyb22@mails.tsinghua.edu.cn

Hong Mei
meih@pku.edu.cn

[1] Department of Computer Science and Technology, Tsinghua University, Beijing, China

[2] Beijing National Research Center for Information Science and Technology, Beijing, China

[3] MoE Key Lab of High Confidence Software Technologies, Peking University, Beijing, China

Journal: 11263 MS: 2413 ☐ TYPESET ☐ DISK ☐ LE ☐ CP Disp.:2025/3/14 Pages: 14 Layout: Large

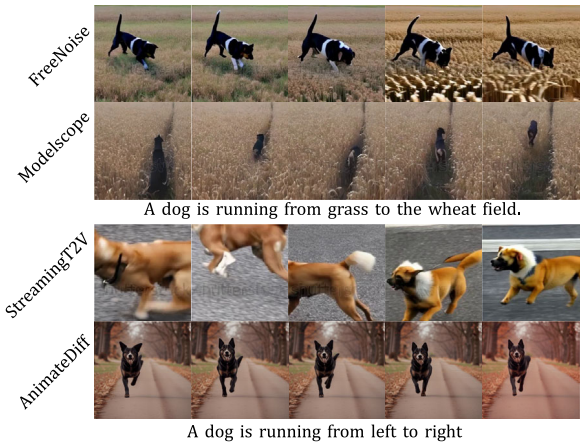


Fig. 1 Some cases with the dynamic scene or subject trajectory changes generated by existing wildly used methods. [29] changes the scene with a transparent gradient manner, which is unnatural. [40] fails to change the scene and makes the background all the same during the entire video. [14] makes the trajectory of the dog unordered and is different from the description of the prompt. [12] generates a running dog but without any position change

of the background should be grass at the beginning of the video. As time moves on, wheat fields gradually appear at the edges of the video. By the end of the video, the background has changed almost entirely to a wheat field, and the dog should be completely in the wheat field. However, existing methods struggle to control the process at this level. The generated videos may exhibit unnatural transformations from grass to the wheat field as shown in the first row of Fig. 1 or the scene changes a little with the background is a mixture of grass and the wheat field shown in the second row of Fig. 1. Similarly, when generating a video with the prompt "*a dog is running from left to right*", existing methods may produce videos with unordered motion patterns shown in the third row. In some cases, the dog even appears to be stationary presented in the last row.

To tackle the weakness of existing works, in this paper, we focus on generating videos with controllable dynamic scene transformations and object trajectories, which faces the following challenges. On the one hand, generating videos with scene transformations requires identifying suitable condition information that enables precise control over object trajectories and dynamic scene transformations. On the other hand, it is hard to incorporate the motion of subjects and the transformations of scenes within short videos. Most of the existing methods [11, 14] try to generate long videos in an auto-regressive manner. However, they ignore the information gap between different scenes, which causes color distortion and results in temporal inconsistency. Thus, presenting these dynamic changes of both scene and object while maintaining video consistency and coherence becomes another big challenge.

To address these challenges, we propose the Dynamic Scene Transformation Text-to-Video Generation model (ScenarioDiff). We utilize a pre-trained text-to-image model, Stable Diffusion, and introduce a temporal module to ensure consistency as the base video generator. To enable the controllable generation of object trajectories and scene transitions, we introduce bounding boxes as the condition information. It is inspired by the practice of previsualization in real-world video production, where simple storyboards are initially created. In order to generate long videos, we propose the mixed frequency controlnet module and use an auto-regressive manner that divides the entire video into several video chunks. To generate each chunk, we utilize the last O frames from the previous chunk as a reference. Additionally, we propose a cross-chunk scheduling mechanism during inference to keep the consistency and smoothness between chunks. Extensive experiments are conducted to show the efficiency of our proposed ScenarioDiff method.

Our contributions are summarized as follows:

- To the best of our knowledge, this work is the first to consider controlling both object motion trajectories and scene transformations in text-to-video generation tasks, making them more consistent with textual descriptions and real-world variations.
- We propose the ScenarioDiff method, which can generate long videos in an auto-regressive manner while maintaining the consistence of the generated videos and satisfying the requirements of conditions.
- Extensive experiments are conducted to demonstrate the superiority of our proposed ScenarioDiff method over existing baselines.

2 Related Works

2.1 Text-to-Image Generation

In recent years, with the development of diffusion models, text-to-image generation has received significant attention and success. Based on large-scale image-text pairs, these models can generate high-quality images that satisfy the text prompts given by the users. [27], as one of the pioneers in diffusion-based text-to-image generation model, introduces classifier-free guidance to achieve better control over text-conditioned image generation. While GLIDE trains a text encoder, [36] utilizes a pre-trained text encoder, allowing it to leverage a wider range of textual data and achieve a deeper understanding of semantics, resulting in improved generation performance. DALL-E [32] also employs a pre-trained text encoder and has subsequently enhanced data quality and improved generation quality by employing higher-quality

Journal: 11263 MS: 2413 TYPESET DISK LE CP Disp.:2025/3/14 Pages: 14 Layout: Large

captions in their subsequent works [3, 31]. The Stable Diffusion series [28, 33] represents a milestone in diffusion-based text-to-image generation models, as it extends the diffusion process to the latent space, ensuring high-quality image outputs while improving the efficiency.

2.2 Text-to-Video Generation

There has been an increasing focus on generative video models due to the success of text-to-image generation models. On the one hand, some approaches [1, 16, 39, 42, 46] directly train on large-scale multi-modal datasets and have achieved promising results. On the other hand, leveraging the success of stable diffusion, some works [5, 12, 13, 18, 20, 23, 37, 40, 45, 49] utilize pre-trained text-to-image generation models and introduce temporal information using different methods, such as cross-frame attention [20] or using an additional temporal module [12], to obtain text-to-video generation models. Notably, OpenAI's Sora [4], which employed transformer architecture to latent diffusion model, can even generate high-quality video with more than 1 min. Despite this progress, general text-to-video methods often fail to faithfully generate the trajectories of the subject described in the text prompt and are struggling to follow the logic of reality and change the background of the video based on the prompt.

2.3 Controllable Generation

Controllable generation aims to generate images or videos with specific conditions. With the development of diffusion methods, more controllable generation methods have been proposed around diffusion. Some methods [11, 17, 48] propose to train a controlnet-branch by duplicating certain parameters of the U-Net architecture to extract conditional information for assisting generation. [10, 26, 47] utilized a new module called adapter to achieve similar effects with a small number of parameters. [21] learns new parameters in attention layers to blend conditions with hidden states for controllable generation. Besides, [44] uses certain gradients of the condition during generation to solve the problem. In addition to controlling the overall image or video, some customization methods [6, 35, 41] employ techniques like Lora to fine-tune specific parameters, allow for localized control and generation of specific aspects of the content, such as the attributes of the subject.

However, existing controllable generation methods for videos are not suitable for the task of this paper since the limitation that most of the methods require corresponding continuous condition information for each frame. This indicates that these methods are more suitable for editing than generating. Besides, obtaining this information becomes inconvenient and difficult as there are serious problems with trying to search for videos of arbitrary object trajectories or scene transformations as required.

3 Methodology

In this section, we will introduce our proposed ScenarioDiff method. The overall framework is shown in Fig. 2, which is built upon the pre-trained AnimateDiff model with Stable Diffusion and GLIGEN as the base model. Thus, in the preliminary, we will provide an overview of Stable Diffusion [12, 33], and [21].

3.1 Preliminary

Stable Diffusion Stable Diffusion has learned relevant knowledge on a large-scale dataset of text-image pairs $\{(x, p)\}$, where x is the image and p is the description of x, enabling it to generate high-quality images based on provided text prompts. The key advantage of Stable Diffusion lies in extending the diffusion process from the pixel level to the latent space utilizing an encoder $\mathcal{E}(\cdot)$ and a decoder $\mathcal{D}(\cdot)$ of [8]. The encoder transforms the image x into the latent code $z_0 = \mathcal{E}(x)$, while the decoder reconstructs the image based on the latent code $x \approx \mathcal{D}(z_0)$. This extension improves efficiency while keeping the quality of the generated images. Specifically, during the diffusion forward process, the Gaussian noise is iteratively added to the latent code as follows:

$$q(z_t|z_{t-1}) = \mathcal{N}(z_t, \sqrt{1-\beta_t}z_{t-1}, \beta_t I), t = 1, 2, ...T, \tag{186}$$

where T is a large number of iterative steps, and the purpose is to transform z_0 into a standard Gaussian noise.

During the diffusion backward stage, Stable Diffusion denoises the Gaussian noise z_T to the latent code z_0 step by step and utilizes a U-Net $\epsilon_\theta(\cdot)$ [34] to predict the noise at each step. At each iteration, the U-Net takes the time step t, the noisy latent z_t, and the encoded text prompt embedding $\tau(p)$ as inputs to predict the noise of the current step, where $\tau(\cdot)$ represents a pre-trained text encoder of [9, 30]. Once the predicted noise is obtained, the diffusion solver is applied to clean the noisy latent. To ensure that the noisy latent still has the information of the text condition during the denoising process, Stable Diffusion employs a cross-attention module as follows:

$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d}}) \cdot V, \tag{201}$$

$$K = W_k \cdot \tau(p), V = W_V \cdot \tau(p), \tag{202}$$

$$Q = W_Q \cdot \phi(z_t), \tag{203}$$

where $\phi(z_t)$ is the representation of noisy latent z_t in the attention layers. (204)

Journal: 11263 MS: 2413 TYPESET DISK LE CP Disp.:2025/3/14 Pages: 14 Layout: Large

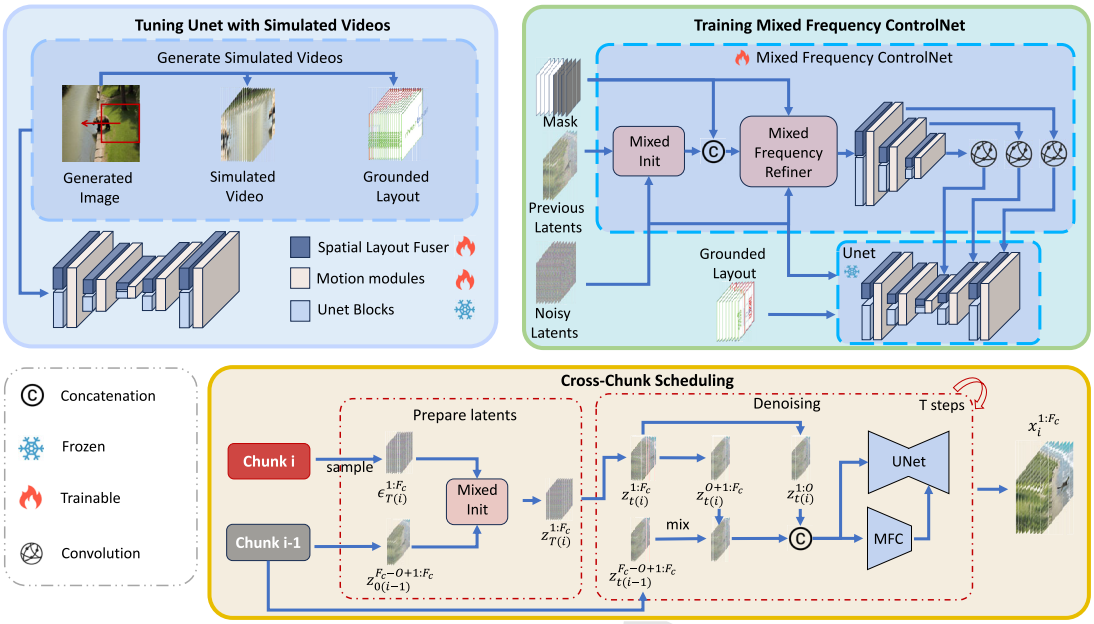


Fig. 2 The framework of ScenarioDiff. Our proposed method contains three main stages. The first stage is tuning the UNet with simulated videos which aims to learn the knowledge of how to change the scenes. The second stage is training the mixed frequency controlnet which allows our method to generate long videos. During the inference stage, we propose a cross-chunk scheduling mechanism that utilizes the information from the previous chunk to make the video more consistent

The optimization objective of Stable Diffusion is defined as follows:

$$min\ \mathbb{E}_{p,z_0,\epsilon,t}[||\epsilon - \epsilon_\theta(z_t, t, \tau(p))||_2^2]$$

where ϵ is a randomly sampled Gaussian noise during the training process, which is added to the latent code through the diffusion forward pass before inputting it into the U-Net. This objective has been widely employed in diffusion-related works.

AnimateDiff AnimateDiff is a text-to-video generation model based on Stable Diffusion. It proposed a Motion Module into the original model to facilitate the transition from images to videos. Specifically, to leverage the capabilities of the existing text-to-image model, AnimateDiff combines the dimension of the length of video frames into the batch size to assist the model in generating high-quality images for each frame. To guarantee the temporal consistency among the generated frames, AnimateDiff introduces a 3D transformer as the Motion Module to learn dynamic information from videos. AnimateDiff is trained on the WebVid dataset and, relying on the prior knowledge from Stable Diffusion, it can generate high-quality and consistent videos. In this paper, we build our ScenarioDiff video generation model based on AnimateDiff.

GLIGEN GLIGEN is a method that tries to control the layout of the image with various grounding information such as bounding box. It is derived from the pre-trained Stable Diffusion with a novel grounding token mechanism and gated self-attention. Suppose there is an object e you want to control and the bounding box of the object is $bbox = [x_1, y_1, x_2, y_2]$, which is the left-top and the right-bottom point of the object and e is the phrase description or image of the object. GLIGEN utilizes a grounding token mechanism to fuse the information as follows:

$$g^e = MLP(\tau(e), Fourier(bbox)),$$

where $MLP(\cdot,\cdot)$ is a multi-layer perceptron and $Fourier$ is the Fourier embedding [25]. To incorporate grounding information into Stable Diffusion, GLIGEN employs a gated self-attention to integrate the obtained grounding token into the hidden states as follows:

$$h = h + \beta \cdot tanh(\gamma) \cdot TS(SelfAttn([h, g^e])),$$

where $TS(\cdot)$ is the operation that selects the dimension of hidden states only, and γ is a learnable scalar that is initialized to 0. Thus, GLIGEN can generate images with the desired layout based on the provided grounding information of objects. In our ScenarioDiff model, we utilize the prior of GLIGEN to control the trajectories of the object.

Task In summary, assuming we have text prompt p and corresponding layout information of each frames $\mathbb{L} = \{L^1, L^2, ..., L^F\}$, where F is the length of video we want to generate. $L^f = \{E^f, B^f\}$ is the layout of the f-th

Journal: 11263 MS: 2413 TYPESET DISK LE CP Disp.:2025/3/14 Pages: 14 Layout: Large

frame, where $E^f = \{e_1^f, e_2^f, ..., e_N^f\}$ is the set of N descriptions of objects or backgrounds to be localized and $B^f = \{bbox_1^f, bbox_2^f, ..., bbox_N^f\}$ is their bounding boxes. The purpose of this paper is to generate videos of length F that satisfy the conditions of the text prompt and layout information using the random sampling noise of $z_t^{1:F} \in \mathbb{R}^{b\times c\times F\times h\times w}$, where b is the batch size, c is the channel of the latent space, F is the number of frames, and h, w are the scaled height and width of the latent space.

3.2 ScenarioDiff

In this section, we provide a detailed description of the components of our proposed ScenarioDiff method. Firstly, we add the spatial layout fuser to the pre-trained AnimateDiff(our base video generator) and train it on a simulated dataset at the first stage. Subsequently, to better generate videos that have the transitions of subject position and scenes, we introduce the mixed frequency controlnet, which assists in generating longer videos in an auto-regressive manner. Lastly, to guarantee the consistency of the videos, we present a cross-chunk scheduling mechanism during inference.

3.2.1 Spatial Layout Fuser

To control the trajectories of the object in each frame of video, we utilize GLIGEN's grounded token mechanism and gated self-attention as the spatial layout fuser and extend it to generate videos. For each block of the Unet in AnimateDiff, we add the spatial layout fuser at the beginning of the cross-attention module.

Before feeding the hidden states into the spatial layout fuser, we reshape the hidden states and combine the temporal dimension of frames with the batch size. This reshaping process enables us to fuse the dynamically changing layout information into the corresponding frames, allowing for controlling the positions of subjects and backgrounds in the video, as described in the preliminary. However, directly applying the GLIGEN modules to the video generator faces the following problem.

As the lack of video datasets that pay attention to dynamic scene transformations during the pretraining of AnimateDiff, even if the input bounding box conditions vary greatly from frame to frame, the resulting scene change is relatively small. To solve the problem, we generate a simulated dataset before we train the model. The process is shown in Fig. 3.

Firstly, we collect a list of common scenes and subjects and randomly pair them up to generate some images that have different scenes on the left and right sides with bounding boxes as conditions. The prompt we used is "*a subject is doing something in a beautiful scene between scene A and scene B, masterpiece, 8K, 4K, high quality, best quality*".

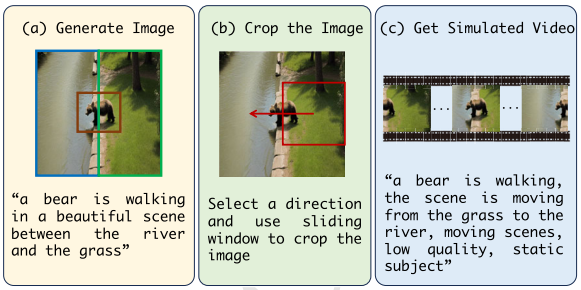


Fig. 3 The process to get simulated videos. We first generate images with bounding boxes as the condition. After that, we use a sliding window to crop the image and give the caption according to the direction

After that, we choose a direction and utilize a sliding window to crop the image as the frames of the video and thus we get a simulated video. However, we acknowledge that the generation process we currently employ is relatively simple. As a result, the simulated data primarily consists of panning movements, lacking dynamic elements, and falling short in terms of data quality.

However, our objective is to capture the knowledge of the transformation of scenes, and we have identified several techniques to mitigate the impact of low-quality data.

- Instead of using a simple prompt like "*A subject is doing something, the scene is moving from scene A to scene B*" as the caption of the video, we enhance it by adding a suffix "*moving scenes, low quality, static subject*" to the prompt. During the inference stage, we just need to describe the scenes we want to change and add the phrase "*moving scenes*" to the prompt.
- As the simulated dataset is low quality and has biases, it is better not to predict the noise ϵ strictly. Thus, we propose a hyperparameter λ_ϵ to introduce a slight intervention as $\hat{\epsilon} = \epsilon + \lambda_\epsilon \epsilon_0$, where ϵ_0 is an additional noise sampled from Gaussian distribution which has the same shape with ϵ.
- To achieve improved performance, we mixed the simulated dataset with a video segmentation dataset that contains temporal layout information.

The final optimization objective for this tine-tuning process is as follows:

$$\mathcal{L}_g = \mathbb{E}_{p^{1:F}, z_0^{1:F}, \epsilon^{1:F}, t}[||\hat{\epsilon} - \epsilon_\theta(z_t^{1:F}, t, \tau(p^{1:F}), \mathbb{L})||_2^2],$$

where $p^{1:F}$ is the text embedding of F frames, $\epsilon_t^{1:F}$ is the noise sampled for F frames, $z_t^{1:F}$ is the noisy latents of F frames at time step t.

Using bounding boxes as conditions for controlling the video generation process offers two key advantages. The first advantage is the ability to conveniently apply conditions to

Journal: 11263 MS: 2413 TYPESET DISK LE CP Disp.:2025/3/14 Pages: 14 Layout: Large

each frame of the generated video, ensuring temporal consistency among the control conditions. This is challenging to achieve with other methods such as canny edge or skeleton when it comes to image generation without a pre-existing video. For instance, to get such bounding box conditions, the user can manually select key points in the motion trajectory and use linear interpolation to obtain the bounding box for each frame. Alternatively, a GUI-based drawing tool can be employed to generate the bounding boxes, or you can even obtain the layout information with the assistance of LLMs. These approaches eliminate the need to search for videos with similar trajectories or scene transformations and extract their canny edges or skeletons. The second advantage lies in the ability to control the layout of the background and set the description or image to the target background, providing greater flexibility.

Thus, we have been able to perform simple control over the trajectories of subjects and the transformations of scenes. However, the current methods are limited to generating short videos of only 16 frames and cannot produce videos with complex position or background transformations that satisfy real-world logic. To address the limitation, we introduce the mixed frequency controlnet to assist in generating longer videos.

3.2.2 Mixed Frequency ControlNet

Due to the limitations in memory and the constraints of the base generator, it is not possible to generate arbitrarily long videos. To generate long videos with F frames, we employ an auto-regressive approach by dividing the entire video into short video chunks as $V_1, V_2, ..., V_I$, where each V_i is a video chunk which has F_c frames. For each chunk, there are O frames of overlapping with the previous chunk and we utilize the overlapping O frames of the previous chunk as reference frames to aid in generating the current chunk. It is evident that $O < F_c$ and only a subset of frames are used as reference frames. However, the controlnet approach only accepts the input with the same shape of the noisy latents as $b \times c \times F_c \times h \times w$. Therefore, the first challenge lies in processing the reference information to ensure that the overlapping frames are nearly identical between adjacent chunks while maintaining temporal consistency for the non-overlapping frames.

To generate the i-th chunk, a naive approach which has been shown in Fig. 4 is to pad frames of zero tensors after the latent code $z_{0(i-1)}^{F_c-O+1:F_c} \in \mathbb{R}^{b\times c\times O\times h\times w}$ of the overlapping frames in the $(i-1)$-th chunk and use the padded code as the conditional input. However, this approach leads to a information gap between the overlapping frames and non-overlapping frames, resulting in a noticeable color distortion at the O-th frame of the final output video chunk and causing inconsistency within the chunk. Besides, for the new chunk

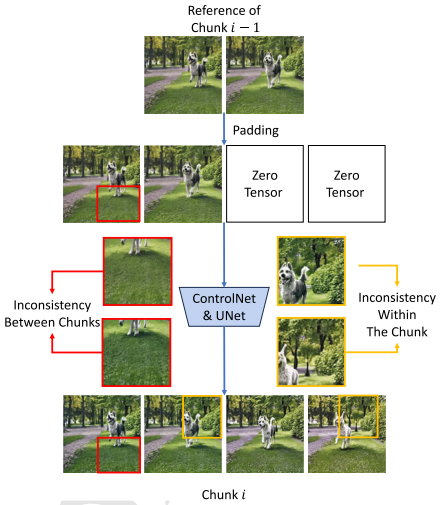


Fig. 4 Naive approach. The first row of images is the reference frames from chunk $i-1$. The second row of images is the reference frames padded with zero tensors. We input the padded information into the model and generate the i-th chunk. However, there are two kinds of inconsistency. The first inconsistency occurs between two chunks, which we mark with a red box. We can find that the generated frames in chunk i are darker than in reference frames. The second inconsistency occurs between the frames with and without reference in chunk i, which we mark with a yellow box. The dog becomes different and the background is brighter

i, the overlapping part generated also exhibits subtle differences, which we have named inconsistency between chunks. We will solve the first inconsistency issue in this section and the second inconsistency issue in the next section.

As the first inconsistency is due to the information gap, one solution is to add some relevant information to the padding tensors. Inspired by [43], the low-frequency components of the latents contain sufficient semantic information and substantial spatio-temporal correlations. Besides, the video generated by the low-frequency latents still keeps high consistency and maintains the style and color. The high-frequency latents pay attention to the motion degree of the video. To keep the consistency of the video while preventing the video from being static suddenly, a more effective approach is to utilize the low-frequency information from the overlapping frames while extracting high-frequency information through the noisy latents of the current timestep. Thus, we proposed a method named Mixed Init, which is illustrated in Fig. 5.

Considering the inherent continuity in videos, we can assume that the latents of non-overlapping frames, denoted as $z_{0(i)}^{f}$, $f \in [O+1, F_c]$, have similar low-frequency information as the latent of the last frame of the overlapping parts $z_{0(i-1)}^{F_c}$. Therefore, we repeat the latent $z_{0(i-1)}^{F_c}$ to serve as an

Journal: 11263 MS: 2413 TYPESET DISK LE CP Disp.:2025/3/14 Pages: 14 Layout: Large

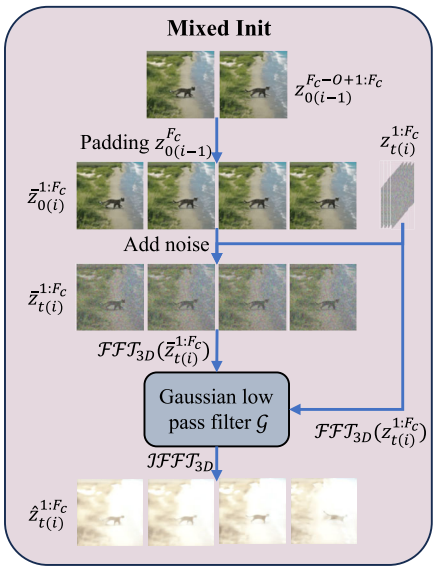


Fig. 5 The process of Mixed Init. The Mixed Init first pads the last frame of reference latents, and then add noise to it. After that, it utilizes a Gaussian low pass filter to mix the information

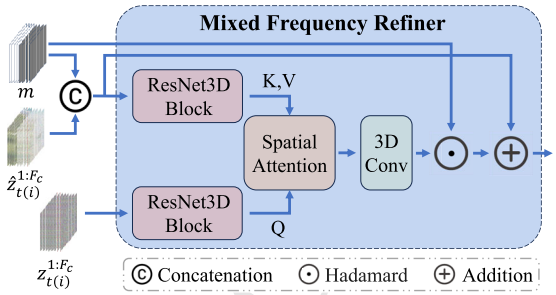


Fig. 6 The framework of proposed mixed frequency refiner module

approximation that make

$$\bar{z}^{f}_{0(i)} = \begin{cases} z^{F_c-O+f}_{0(i-1)}, f \in [1, O], \\ z^{F_c}_{0(i-1)}, f \in [O+1, F_c]. \end{cases}$$

To obtain the reference information of each timestep t, we begin by performing a mixed initialization which uses the fast Fourier transform (FFT) and the Gaussian low pass filter $\mathcal{G}$ to mix the low-frequency and high-frequency information.

$$F^{L}_{\bar{z}^{1:F_c}_{t(i)}} = \mathcal{FFT}_{3D}(\bar{z}^{1:F_c}_{t(i)}) \odot \mathcal{G},$$
$$F^{H}_{z^{1:F_c}_{t(i)}} = \mathcal{FFT}_{3D}(z^{1:F_c}_{t(i)}) \odot (1-\mathcal{G}),$$
$$\hat{z}^{1:F_c}_{t(i)} = \mathcal{IFFT}_{3D}(F^{L}_{\bar{z}^{1:F_c}_{t(i)}} + F^{H}_{z^{1:F_c}_{t(i)}}),$$

where $z^{1:F_c}_{t(i)}$ is the noisy latents of timestep t, and $\bar{z}^{1:F}_{t(i)}$ is the latents extracted from the t-step diffusion forward process performed by $\bar{z}^{1:F}_{0(i)}$. After that, to keep all the original reference information, for each $f \in [1, O]$, we make $\hat{z}^{f}_{t(i)} = \bar{z}^{f}_{0(i)}$. Thus, we get the reference latent $\hat{z}^{1:F_c}_{t(i)}$ of timestep t.

Since we approximate the low-frequency information of $z^{O}_{0(i)}$ in place of the ground truth low-frequency information of the subsequent frames, the obtained reference $\hat{z}^{1:F_c}_{t(i)}$ is not perfect. To better improve the result, we add a slight perturbation $\Delta\hat{z}^{f}_{t(i)}$ to each frame in the non-overlapping part. We propose a network called mixed frequency refiner to learn $\Delta\hat{z}^{f}_{t(i)}$ and the framework is shown as Fig. 6.

The mixed frequency refiner takes the reference information $\hat{z}^{1:F_c}_{t(i)}$ and the noisy latent $z^{1:F_c}_{t(i)}$ of the current timestep as inputs. To distinguish between the overlapping and non-overlapping parts, we introduce a mask $m \in \mathbb{R}^{b\times 1\times F_c\times h\times w}$, where it takes the value of 1 for the overlapping frames and 0 for the non-overlapping frames. We concatenate this mask with the reference latent and get $\hat{z}^{1:F_c}_{t(i)} = [\hat{z}^{1:F_c}_{t(i)}, m] \in \mathbb{R}^{b\times(c+1)\times F_c\times h\times w}$. Then, we use a ResNet block with learnable time embedding to obtain hidden states of reference $h_{ref}(\hat{z}^{1:F_c}_{t(i)}, t)$ as follows:

$$h = conv(silu(norm(\hat{z}^{1:F_c}_{t(i)}))),$$
$$temb = \mathcal{T}_{\theta}(silu(t)),$$
$$h = conv(silu(norm(h + temb))),$$
$$h_{ref}(\hat{z}^{1:F_c}_{t(i)}, t) = \hat{z}^{1:F_c}_{t(i)} + h,$$

where $silu$ is SiLU function [7] and $\mathcal{T}_{\theta}(\cdot)$ is a learnable time embedding. We use another ResNet block with the same architecture to get the hidden states of noisy latents $h(z^{1:F_c}_{t(i)}, t)$.

Subsequently, we treat the hidden states of reference latents as key and value, while the hidden states of noisy latents serve as query. By employing attention mechanisms, we aim to capture the relationship between the current latents and reference information, expecting to obtain corresponding slight perturbations $\Delta\hat{z}^{f}_{t(i)}$ as

$$\Delta\hat{z}^{f}_{t(i)} = Softmax(\frac{QK^{T}}{\sqrt{d}}) \cdot V,$$
$$K = W_k \cdot h_{ref}(\hat{z}^{1:F_c}_{0(i)}, t),$$
$$V = W_V \cdot h_{ref}(\hat{z}^{1:F_c}_{0(i)}, t),$$
$$Q = W_Q \cdot h(z^{1:F_c}_{t(i)}).$$

As we only need to add slight perturbations to the non-overlapping frames, we utilize the previously mentioned mask to obtain a masked input $w = \hat{z}^{1:F_c}_{t(i)} + (1-m) \cdot conv(\Delta\hat{z}^{1:F_c}_{t(i)})$, which is then fed into the controlnet branch.

Since there is no ground truth of $\Delta\hat{z}^{1:F_c}_{t(i)}$, treating the information of overlapping frames and the processed reference equally would result in a decline in quality. Hence, we apply

Journal: 11263 MS: 2413 TYPESET DISK LE CP Disp.:2025/3/14 Pages: 14 Layout: Large

weighting to the optimization objective to address this issue as follows:

$$\mathcal{L} = \frac{1}{F_c}\sum_{f=1}^{F_c} \lambda_f \cdot \mathbb{E}_{p^f, z^f, \epsilon^f, t}[||\epsilon^f - \epsilon_\theta(z^f, t, \tau(p^f))||_2^2],$$

where $\lambda_f = 1$ for $f \leq O$ and $\lambda_f = 1 - \frac{(1-\hat{\lambda})(f-O)}{F_c - O}$ for $f > O$. $\hat{\lambda} \in [0, 1]$ is a hyperparameter of minimum weight.

Simultaneously, when memory is limited, it is recommended to employ the strategy of continuous batch training instead of using simply b different video crops as a data batch. Specifically, for the training samples, a video exceeding F frames is collected, which is then divided into b chunks with overlapping segments, forming a batch for training. It can better alleviate the potential possibility of color distortion issues when utilizing controlnet for auto-regressive generation.

3.2.3 Cross-chunk Scheduling

After training, the continuity within each chunk is highly guaranteed. However, the introduction of reference information has led to slight distributional changes, and the continuity between chunks still needs to be addressed. Directly selecting results from a single chunk would lead to overall video discontinuity. Therefore, we address the continuity of the whole video by scheduling the latents from two aspects during inference.

The first aspect pertains to the latents sampling. We adopt the approach of Mixed Init. The second aspect is scheduling the latents of the overlapping frames during the denoising process. To enhance the continuity between different chunks, we store and reuse the latents for each timestep of the overlapping frames from the previous chunk. When generating the current chunk, we selectively replace the results of the current chunk with the latents from the previous chunk, based on a specific ratio. Intuitively, to achieve smooth transitions between the video chunks, frames closer to the previous chunk should be similar to the stored latents, while frames closer to the subsequent frames should utilize the latents obtained by denoising the current noisy latents. Therefore, we employ a simple interpolation technique to improve the continuity of the video as

$$z^f_{t(i)} = (1 - \frac{f}{O}) \cdot z^f_{t(i-1)} + \frac{f}{O} \cdot z^f_{t(i)}, \forall f \in [1, O].$$

While for non-overlapping frames, we keep the value of $z^f_{t(i)}$.

After finishing denoising, we get pixel-level video $x_i^{1:F_c} \in \mathbb{R}^{b\times c\times f\times H\times W}$ by decoding the latents. In order to further ensure the consistency of the video between different chunks, there is an optional operation which we perform pixel-level normalization on the video after decoding. For each frame f in the overlapping part, i.e. $f \in [1, O]$, we make the mean $\mu^f_{i,j}$ and the standard deviation $\sigma^f_{i,j}$ of each channel j of the current chunk i the same as the corresponding frame of the previous chunk as follows:

$$x^f_{i,j} = \frac{x^f_{i,j} - \mu^f_{i,j}}{\sigma^f_{i,j}} * \sigma^{F_c-O+f}_{i-1,j} + \mu^{F_c-O+f}_{i-1,j}.$$

For the frame in the non-overlapping part, i.e. $f \in [O + 1, F_c]$, we take the mean and the standard deviation of the last frame of the overlapping part as reference.

$$x^f_{i,j} = \frac{x^f_{i,j} - \mu^f_{i,j}}{\sigma^f_{i,j}} * \sigma^{F_c}_{i-1,j} + \mu^{F_c}_{i-1,j}.$$

4 Experiments

4.1 Experiment Settings

The ScenarioDiff method is built upon AnimateDiff v2 and Stable Diffusion v1.5. Our method consists of two training stages. In the first stage, we jointly train the spatial layout fuser and the motion module on the simulated dataset and the VSPW dataset [24]. During this stage, we use a learning rate of 1e-5, the weight λ_ϵ of 0.05, and a batch size of 2. In the second stage, we train the mixed frequency controlnet on a subset of WebVid dataset [2]. In this stage, we use a learning rate of 1e-5, the weight $\hat{\lambda}$ of 0.8, and a batch size of 2. Throughout all stages, the video data used is sampled into 16 frames. All training was performed on a single NVIDIA A100 GPU.

For evaluation, we select several common objects such as dogs, cats, and humans, and some common scenes. We then provide them with a specified motion direction or some transformation rules, resulting in prompts such as "a dog is running from left to right" or "a person is walking from a beach to a wheat field". Ultimately, we generated 350 videos for each method for comparison. For all videos, we set the length of video chunk $F_c = 16$, with the overlapping length of $O = 8$.

Baselines To better evaluate the performance of ScenarioDiff, we compared it against various baselines. Their basic information is provided below:

- [40]. Modelscope introduces spatial-temporal convolution and attention mechanisms into the [33] framework enabling the generation of high-quality videos.

Journal: 11263 MS: 2413 TYPESET DISK LE CP Disp.:2025/3/14 Pages: 14 Layout: Large

Table 1 The quantitative results of the baselines and our proposed ScenarioDiff method. The bolded results represent the optimal results, and the underlined results mean the sub-optimal results

Methods	Consistency	Smoothness	Dynamic	CLIP-sim	User-rank
Modelscope	0.2181	0.9616	0.6809	0.2346	2.7481
SparseCtrl	0.2161	0.9684	0.6814	0.2262	3.6333
StreamingT2V	0.2127	0.9622	0.6686	0.2316	3.8000
FreeNoise	0.2205	0.9690	0.4400	**0.2471**	3.5519
Ours	**0.2293**	**0.9744**	**0.6857**	0.2346	**1.2667**

- [11]. SparseCtrl is a controlnet-based method to control the generation of videos based on several types of condition information built upon [12].
- [14]. StreamingT2V is a controlnet-based method that uses an auto-regressive manner to generate long videos. To get more consistent chunks, StreamingT2V proposes a randomized blending approach to mixture latents between chunks.
- [29]. FreeNoise is a tuning-free method to generate long videos while preserving consistency. To achieve this, FreeNoise reschedules the noises and performs attention over them by window-based fusion.

Metrics In order to evaluate the performance of the methods from multiple perspectives, we employed diverse metrics to assess these approaches as follows:

- [19]. VBench is a benchmark for video generation that designs several metrics to evaluate the quality of video in various aspects. We utilize some of these metrics which have high relations to our setting including overall consistency, motion smoothness, and dynamic degree.
- CLIP-Sim. To measure the alignment between generated videos and provided text prompts, we utilized CLIP to compare the similarity of each frame in the generated videos with the corresponding text and computed the average of these similarities as the final score.
- User-Rank. In order to evaluate the methods more accurately, we conducted a user study. We presented users with several prompts and videos generated by the methods and asked them to rank the videos based on how well they aligned with the text description.

4.2 Main Results

The scores of different methods are shown in Table 1, and some visualized generated results are shown in Fig. 7.

The quantitative comparisons demonstrate that our proposed method achieves better performance than all the baselines. Our method gets sub-optimal results only in terms of the CLIP-sim metric while outperforming all other metrics. This indicates that our proposed method ensures the generated videos are semantically consistent while possessing stronger smoothness and dynamics. FreeNoise performs well in most metrics but does poorly in terms of dynamic degree. This may be attributed to its reuse of noisy latents without introducing new dynamic information, constraining the entire scene within a limited space.

Next, we will analyze the qualitative results. Figure 7 showcases the generated videos based on parts of the prompts. In the case of the prompt "a dog is running from the grass to the beach", requires generating videos with a scene transition. Most baselines blend the information from both "the grass" and "the beach" throughout the entire video, lacking noticeable changes in the overall video scene. In contrast, our generated video begins with a dog standing on the grass, facing the beach. As the video progresses, the dog gradually runs towards the beach, and the entire video scene moves completely to the beach. This is the same as the semantic of the prompt while satisfying the logic of the real-world scene changing.

In (b), the objective is to control the trajectory of the subject from right to left. We observe several issues with the baselines.

- Random subject orientation. It means the baselines struggle to effectively control the trajectory, sometimes even resulting in movements in the opposite direction, as seen in the results generated by StreamingT2V.
- Insufficient object motion. Both Modelscope and FreeNoise generate videos where the subject is facing the left, but the relative positions hardly change throughout the video.

Our result successfully depicts the subject running from the right side of the video to the left side, satisfying the prompts.

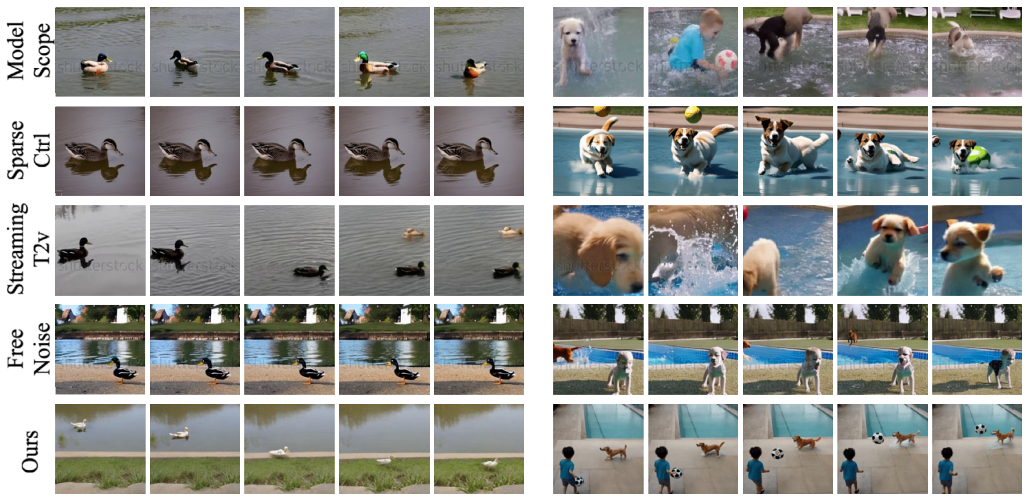
Rather than only controlling the trajectory, there are additional requirements regarding the background, the trajectory, and the subject's interaction with these backgrounds over time in (c). We observe that only FreeNoise and our method generate both a pool and the land to accommodate the text "walks ashore". However, FreeNoise fails to consider "the duck is swimming", resulting in unsatisfactory results.

(d) attempts to control the trajectories of multiple objects. It can be observed that as the control requirements increase, other baselines somehow overlook parts of the information.

Journal: 11263 MS: 2413 TYPESET DISK LE CP Disp.:2025/3/14 Pages: 14 Layout: Large

(a) A dog is running from grass to beach.

(b) A person is riding a horse under the mountain from right to left.



(c) A duck is swimming in the pool and then walks ashore.

(d) A boy kicks a ball into a pool of water and a dog runs over to it.

Fig. 7 The prompts and results generated by the baselines and our method

For example, the video generated by Modelscope appears chaotic and fails to convey logical coherence. SparseCtrl ignores the prompt of "a boy kicks a ball" and only generates a dog running over a ball. In contrast, our method achieves satisfactory results by incorporating scene information.

4.3 Ablation Study

In this section, we conducted a detailed analysis of some of the hyperparameters and different components of our proposed method.

4.3.1 Tuning with Simulated Dataset

We have experimented to analyze the effectiveness of tuning on the simulated dataset, which is the first stage of our proposed method. The results are shown in Fig. 8.

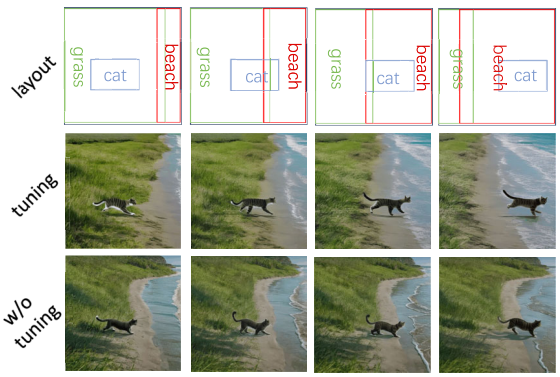


Fig. 8 Qualitative results when ScenarioDiff with and without tuning on the simulated dataset. The first row is the layout provided when generation. The second row is the generated result with tuning on the simulated dataset. The third row is the generated result without tuning on the simulated dataset

Journal: 11263 MS: 2413 TYPESET DISK LE CP Disp.:2025/3/14 Pages: 14 Layout: Large

Table 2 Ablations about the components of the mixed frequency controlnet

	Consistency	Smoothness	CLIP-sim
w/o both	0.1584	0.9642	0.2190
w/o mixed	0.1540	0.9759	0.2206
w/o refiner	0.1733	**0.9842**	0.2284
ours	**0.1748**	0.9837	**0.2304**

Fig. 9 Qualitative results when ScenarioDiff with and without the components of mixed frequency controlnet. The video chunk is generated with the prompt "A dog is running on the park". The first two columns of images are sampled from the reference frames, and the last two columns are predicted based on the reference

It can be observed that without fine-tuning, the model is only able to control the motion of subjects but cannot control scene movement and transformation. This is because the pre-training data consists mainly of short video clips focused on the motion of subjects, and lacks sensitivity to background changes. After introducing the simulated dataset with varying backgrounds, this issue is greatly alleviated, allowing for effective control of scene transformations through bounding boxes.

4.3.2 Mixed Frequency ControlNet

In mixed frequency controlnet, there are two main components, mixed init mechanism and mixed frequency refiner. We have done experiments to evaluate the model without each of the components to show their effectiveness. The results are shown in Table 2 and some of the generated videos are shown in Fig. 9.

Considering the quantitative metrics, the removal of each of the modules results in a decrease in the overall generative ability of the model. Specifically, when the mixed init mechanism is removed, there is a noticeable decline in three metrics. This suggests that the utilization of the mixed init to expand and mix the low-frequency information from the reference video chunk indeed assists the model in generating smooth video frames while maintaining consistency with the reference video frames.

The qualitative results also support our point. For the model without mixed init and mixed frequency refiner, it still can generate videos in an auto-regressive manner. However, there exists a problem that the color and the quality changes a lot within the chunk with and without reference. In the first row, the dogs in the third and the fourth columns are much different than in the references, and the grass in the background is not completely as green as in the first columns but has some yellowing. When we remove only the mixed init, it is even worse in some cases. This is because the mixed frequency refiner is used to add perturbations to the part of the non-overlapping frames to estimate the mixing frequency information of the subsequent frames and to improve the quality of the generation after the mixed init is done. If mixed init is missing, the meaning of this module is confusing. When only removing the mixed frequency refiner, without additional processing, we give the non-overlapping part almost the same reference information as the last frame of the overlapping part. It will mislead the model and conflict with other conditioning information which will finally cause a decrease in performance.

Table 3 Ablations about the influence of the hyperparameter $\hat{\lambda}$

	Consistency	Smoothness	CLIP-sim
$\hat{\lambda} = 0.00$	0.1718	0.9778	0.2225
$\hat{\lambda} = 0.25$	0.1739	0.9799	0.2299
$\hat{\lambda} = 0.50$	0.1765	0.9812	0.2295
$\hat{\lambda} = 0.75$	0.1749	0.9821	0.2338
$\hat{\lambda} = 1.00$	0.1744	0.9828	0.2303

4.3.3 Hyperparameter

In the previous subsection, we illustrate the effectiveness of the framework of the mixed frequency controlnet, and in this subsection, we try to discuss the hyperparameter used for its training. We conduct experiments on the parameter $\hat{\lambda}$. We have trained the model with different values of $\hat{\lambda}$ and evaluated the performances on a subset of prompts used in the main experiment. The results are shown in Table 3 and Fig. 10.

From the quantitative results, both consistency and CLIP-sim increase and decrease as $\hat{\lambda}$ increases. When $\hat{\lambda} = 1$, it degrades to train the controlnet normally, which indicates that our proposed method is effective.

Journal: 11263 MS: 2413 TYPESET DISK LE CP Disp.:2025/3/14 Pages: 14 Layout: Large

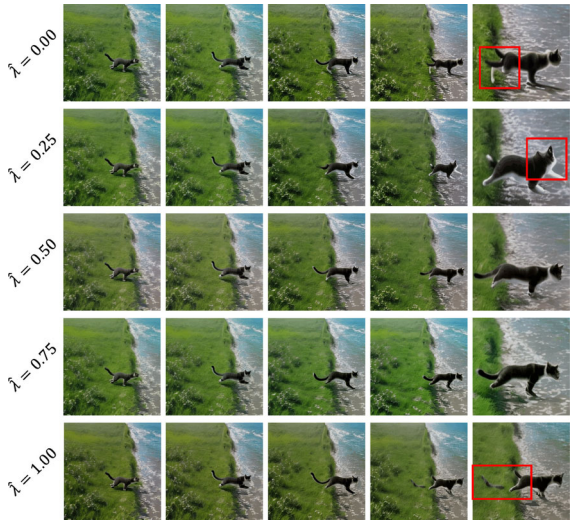


Fig. 10 Qualitative results of different $\hat{\lambda}$. The video chunk is generated with the prompt "A cat is running from the grass to the beach". We have enlarged the cat of the fourth column in the last column and highlighted with a red box to point the degradation in quality that may occur when $\hat{\lambda}$ is too large or too small

Table 4 Ablations about the cross-chunk scheduling

	Consistency	Smoothness	CLIP-sim
w/o both	0.1727	0.9795	0.2241
w/o mixed	0.1744	**0.9843**	0.2249
w/o reuse	0.1730	0.9811	0.2291
ours	**0.1748**	0.9837	**0.2304**

From the qualitative results, we notice that the quality of the cats in the non-overlapping part drops severely when $\hat{\lambda}$ is small. This may be because the information on the non-overlapping part estimated by the mixed init and the mixed frequency refiner is not utilized, and the non-overlapping part is seldom reconstructed during the training stage. When $\hat{\lambda}$ is set to a large value, the stacking of low-frequency information from the last frame of the overlapping part may result in residual artifacts in the generated video, as observed in the cat's tail in the last column of the last raw in Fig. 10. Therefore, it is crucial to appropriately select the value of $\hat{\lambda}$.

4.3.4 Cross-Chunk Scheduling

In the subsection, we make a discussion about the mixed init mechanism and the latents reuse mechanism utilized during cross-chunk scheduling. The quantitative results are shown in Table 4 and the qualitative videos are shown in Fig. 11.

It can be found that when there is no reuse latents mechanism, the video will have color distortion between chunks. For example, the street's color will become yellow after changing chunks in the first row and the fourth column in

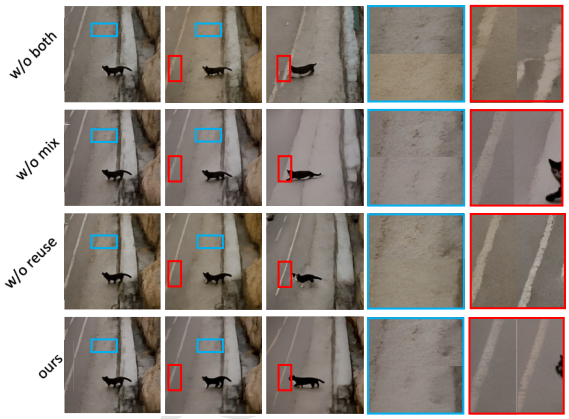


Fig. 11 Qualitative results when ScenarioDiff with and without the components of cross-chunk scheduling. The video chunk is generated with the prompt "A cat is running from the rock to the street". The first three columns are sampled from different video chunks. We highlight regions that may exhibit inconsistency between different chunks and mark them with boxes. In the last two columns, we enlarge these regions and mark them with boxes of the same color

Fig. 11. In the third row, the color of the lines on the street also varied as we can find in the last column, resulting in inconsistency. Besides, when not using mixed init, the video generated will lose information and get a low-quality result. As shown in the second row of Fig. 11, in the last column, the generated video forgets the lane lines on the street.

5 Discussion

The limitations of the paper lie in two aspects. The first aspect lies in the fact that this method inherits from AnimateDiff and GLIGEN and therefore inherits their limitations. The second limitation is the scarcity and low quality of video data with bounding boxes. Because of the difficulty of labeling, the scales of datasets for video segmentation are currently small. Besides, the simulated videos generated from images are low quality.

During this work, we found some biases, conflicts, and compromises among different conditioning information. For example, as shown in Fig. 12, we want to generate videos of "a person is standing on the horseback". The bias of text conditions lies in that most of the relations of person and horse in the pre-trained datasets are "riding". As a result, with only text prompt, the person is always sitting on the horse. With well-designed layouts, we can generate a person standing on the horse. However, the video quality and the degree to which the objects match the bounding boxes are reduced. This implies that there exist conflicts if we use multiple different condition information.

Therefore, a potential direction for future research is to learn the relationships among a broader range of condition-

Journal: 11263 MS: 2413 TYPESET DISK LE CP Disp.:2025/3/14 Pages: 14 Layout: Large

Fig. 12 The comparison of generated videos of prompt "a person is standing on the horseback" with and without the layout. The first row is the video without the layout. The second row is the video with the layout. The last row is the layout provided

ing information, aiming to discover a balance that leads to superior controllable generation.

6 Conclusion

In this paper, we present a novel text-to-video generation framework ScenarioDiff. ScenarioDiff can generate videos that transfer the scenes satisfying real-world logic, with the proposed spatial layout fuser, mixed frequency controlnet, and cross-chunk scheduling mechanism. Extensive experiments on ScenarioDiff demonstrate the effectiveness of the proposed method.

Funding This work is supported by the National Key Research and Development Program of China No.2023YFF1205001, National Natural Science Foundation of China (No. 62222209, 62250008, 62102222), Beijing National Research Center for Information Science and Technology under Grant No. BNR2023RC01003, BNR2023TD03006, and Beijing Key Lab of Networked Multimedia.

Data Availability This paper uses public datasets to conduct experiments available in the following URLs. [24]: https://www.vspwdataset.com/. [2]: https://github.com/m-bain/webvid.

References

1. Bain, M., Nagrani, A., Varol, G., & Zisserman, A. (2021a). 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 (pp. 1728–1738).
2. Bain, M., Nagrani, A., Varol, G., & Zisserman, A. (2021b). 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 (pp. 1728–1738).
3. Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., & Li, L. others (2023). Improving image generation with better captions. Computer Science. https://cdn.openai.com/papers/dall-e-3.pdf, 2(3), 8,
4. Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., & Ramesh, A. (2024). Video generation models as world simulators. https://openai.com/research/video-generation-models-as-world-simulators
5. Chen, H., Xia, M., He, Y., Zhang, Y., Cun, X., & Yang, S. others (2023). Videocrafter1: Open diffusion models for high-quality video generation. arXiv preprint arXiv:2310.19512
6. Chen, H., Zhang, Y., Wu, S., Wang, X., Duan, X., Zhou, Y., & Zhu, W. (2023). Disenbooth: Identity-preserving disentangled tuning for subject-driven text-to-image generation. The twelfth international conference on learning representations.
7. Elfwing, S., Uchibe, E., & Doya, K. (2018). Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks, 107*, 3–11.
8. Esser, P., Rombach, R., & Ommer, B. (2021). Taming transformers for high-resolution image synthesis. In: Proceedings of the ieee/cvf conference on computer vision and pattern recognition (pp. 12873–12883).
9. Frans, K., Soros, L., & Witkowski, O. (2022). Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *Advances in Neural Information Processing Systems, 35*, 5207–5218.
10. Guo, X., Zheng, M., Hou, L., Gao, Y., Deng, Y., & Ma, C. others (2023). I2v-adapter: A general image-to-video adapter for video diffusion models. arXiv preprint arXiv:2312.16693
11. Guo, Y., Yang, C., Rao, A., Agrawala, M., Lin, D., & Dai, B. (2023). Sparsectrl: Adding sparse controls to text-to-video diffusion models. arXiv preprint arXiv:2311.16933
12. Guo, Y., Yang, C., Rao, A., Liang, Z., Wang, Y., Qiao, Y., & Dai, B. (2023). Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. The twelfth international conference on learning representations.
13. He, Y., Yang, T., Zhang, Y., Shan, Y., & Chen, Q. (2022). Latent video diffusion models for high-fidelity long video generation. arXiv preprint arXiv:2211.13221
14. Henschel, R., Khachatryan, L., Hayrapetyan, D., Poghosyan, H., Tadevosyan, V., Wang, Z. & Shi, H. (2024). Streamingt2v: Consistent, dynamic, and extendable long video generation from text. arXiv preprint arXiv:2403.14773
15. Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems, 33*, 6840–6851.
16. Hong, W., Ding, M., Zheng, W., Liu, X., & Tang, J. (2022). Cogvideo: Large-scale pretraining for text-to-video generation via transformers. The eleventh international conference on learning representations.
17. Hu, Z., & Xu, D. (2023). Videocontrolnet: A motion-guided video-to-video translation framework by using diffusion model with controlnet. arXiv preprint arXiv:2307.14073
18. Huang, H., Feng, Y., Shi, C., Xu, L., Yu, J., & Yang, S. (2024). Freebloom: Zero-shot text-to-video generator with llm director and ldm animator. Advances in Neural Information Processing Systems, 36
19. Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y. & Liu, Z. (2024). VBench: Comprehensive benchmark suite for video generative models. Proceedings of the ieee/cvf conference on computer vision and pattern recognition.
20. Khachatryan, L., Movsisyan, A., Tadevosyan, V., Henschel, R., Wang, Z., Navasardyan, S., & Shi, H. (2023). Text2video-zero: Text-to-image diffusion models are zero-shot video generators. Proceedings of the ieee/cvf international conference on computer vision (pp. 15954–15964).
21. Li, Y., Liu, H., Wu, Q., Mu, F., Yang, J., Gao, J., & Lee, Y.J. (2023). Gligen: Open-set grounded text-to-image generation. In: Proceedings of the ieee/cvf conference on computer vision and pattern recognition (pp. 22511–22521).
22. Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., & Zhu, J. (2022). Dpm-solver: A fast ode solver for diffusion probabilistic model sampling

Journal: **11263** MS: **2413** ☐ TYPESET ☐ DISK ☐ LE ☐ CP Disp.:**2025/3/14** Pages: **14** Layout: **Large**

in around 10 steps. *Advances in Neural Information Processing Systems, 35*, 5775–5787.
23. Luo, Z., Chen, D., Zhang, Y., Huang, Y., Wang, L., Shen, Y., & Tan, T. (2023). Videofusion: Decomposed diffusion models for high-quality video generation. In: Proceedings of the ieee/cvf conference on computer vision and pattern recognition (pp. 10209–10218).
24. Miao, J., Wei, Y., Wu, Y., Liang, C., Li, G., & Yang, Y. (2021). Vspw: A large-scale dataset for video scene parsing in the wild. In: Proceedings of the ieee/cvf conference on computer vision and pattern recognition (pp. 4133–4143).
25. Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2021). Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM, 65*(1), 99–106.
26. Mou, C., Wang, X., Xie, L., Wu, Y., Zhang, J., Qi, Z., & Shan, Y. (2024). T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In: Proceedings of the aaai conference on artificial intelligence (Vol. 38, pp. 4296–4304).
27. Nichol, A.Q., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., Mcgrew, B., & Chen, M. (2022). Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In: International conference on machine learning (pp. 16784–16804).
28. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., & Rombach, R. (2023). Sdxl: Improving latent diffusion models for high-resolution image synthesis. The twelfth international conference on learning representations.
29. Qiu, H., Xia, M., Zhang, Y., He, Y., Wang, X., Shan, Y., & Liu, Z. (2023). Freenoise: Tuning-free longer video diffusion via noise rescheduling. arXiv preprint arXiv:2310.15169, , ,
30. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., & Agarwal, S. others (2021). Learning transferable visual models from natural language supervision. International conference on machine learning (pp. 8748–8763).
31. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125, 1(2), 3
32. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., & Sutskever, I. (2021). Zero-shot text-to-image generation. International conference on machine learning (pp. 8821–8831).
33. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In: Proceedings of the ieee/cvf conference on computer vision and pattern recognition (pp. 10684–10695).
34. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. Medical image computing and computer-assisted intervention–miccai 2015: 18th international conference, munich, germany, october 5-9, 2015, proceedings, part iii 18 (pp. 234–241).
35. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., & 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 (pp. 22500–22510).
36. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., et al. (2022). Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems, 35*, 36479–36494.
37. Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., & Zhang, S. others (2022). Make-a-video: Text-to-video generation without text-video data. The eleventh international conference on learning representations.
38. Song, J., Meng, C., & Ermon, S. (2020). Denoising diffusion implicit models. International conference on learning representations.
39. Villegas, R., Babaeizadeh, M., Kindermans, P., J., Moraldo, H., Zhang, H., Saffar, M.T., & Erhan, D. (2022). Phenaki: Variable length video generation from open domain textual descriptions. International conference on learning representations.
40. Wang, J., Yuan, H., Chen, D., Zhang, Y., Wang, X., & Zhang, S. (2023). Modelscope text-to-video technical report. arXiv preprint arXiv:2308.06571
41. Wang, Z., Li, A., Xie, E., Zhu, L., Guo, Y., Dou, Q., & Li, Z. (2024). Customvideo: Customizing text-to-video generation with multiple subjects. arXiv preprint arXiv:2401.09962
42. Wu, C., Huang, L., Zhang, Q., Li, B., Ji, L., Yang, F., & Duan, N. (2021). Godiva: Generating open-domain videos from natural descriptions. arXiv preprint arXiv:2104.14806
43. Wu, T., Si, C., Jiang, Y., Huang, Z., & Liu, Z. (2023). Freeinit: Bridging initialization gap in video diffusion models. arXiv preprint arXiv:2312.07537
44. Xie, J., Li, Y., Huang, Y., Liu, H., Zhang, W., Zheng, Y., & Shou, M.Z. (2023). Boxdiff: Text-to-image synthesis with training-free box-constrained diffusion. Proceedings of the ieee/cvf international conference on computer vision (pp. 7452–7461).
45. Xing, J., Xia, M., Liu, Y., Zhang, Y., He, Y., & Liu, H. others (2024). Make-your-video: Customized video generation using textual and structural guidance. IEEE Transactions on Visualization and Computer Graphics
46. Xue, H., Hang, T., Zeng, Y., Sun, Y., Liu, B., Yang, H., & Guo, B. (2022). Advancing high-resolution video-language representation with large-scale video transcriptions. In: Proceedings of the ieee/cvf conference on computer vision and pattern recognition (pp. 5036–5045).
47. Ye, H., Zhang, J., Liu, S., Han, X., & Yang, W. (2023). Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. arXiv preprint arXiv:2308.06721
48. Zhang, L., Rao, A., & Agrawala, M. (2023). Adding conditional control to text-to-image diffusion models. In: Proceedings of the ieee/cvf international conference on computer vision (pp. 3836–3847).
49. Zhou, D., Wang, W., Yan, H., Lv, W., Zhu, Y., & Feng, J. (2022). Magicvideo: Efficient video generation with latent diffusion models. arXiv preprint arXiv:2211.11018