Vlogger: Make Your Dream A Vlog

Shaobin Zhuang[1,2♠] Kunchang Li[3,4,2♠] Xinyuan Chen[2♡]
Yaohui Wang[2♡] Ziwei Liu[5] Yu Qiao[2] Yali Wang[3,2♡]

[1]Shanghai Jiao Tong University [2]Shanghai AI Laboratory

[3]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences

[4]University of Chinese Academy of Sciences [5]S-Lab, Nanyang Technological University

Abstract

*In this work, we present **Vlogger**, a generic AI system for generating a **minute**-level video blog (i.e., vlog) of user descriptions. Different from short videos with a few seconds, vlog often contains a complex storyline with diversified scenes, which is challenging for most existing video generation approaches. To break through this bottleneck, our Vlogger smartly leverages Large Language Model (LLM) as Director and decomposes a long video generation task of vlog into four key stages, where we invoke various foundation models to play the critical roles of vlog professionals, including (1) Script, (2) Actor, (3) ShowMaker, and (4) Voicer. With such a design of mimicking human beings, our Vlogger can generate vlogs through explainable cooperation of top-down planning and bottom-up shooting. Moreover, we introduce a novel video diffusion model, **ShowMaker**, which serves as a videographer in our Vlogger for generating the video snippet of each shooting scene. By incorporating Script and Actor attentively as textual and visual prompts, it can effectively enhance spatial-temporal coherence in the snippet. Besides, we design a concise mixed training paradigm for ShowMaker, boosting its capacity for both T2V generation and prediction. Finally, the extensive experiments show that our method achieves state-of-the-art performance on zero-shot T2V generation and prediction tasks. More importantly, Vlogger can generate over 5-minute vlogs from open-world descriptions, without loss of video coherence on script and actor.*

1. Introduction

Vlogs represent a unique form of blogging, distinguished by their utilization of video as the primary medium rather than text. Due to more lively expression in the dynamic scenes, vlog has become one of the most popular online-sharing ways in the digital world. In the past two years, the remarkable success of diffusion models [31, 32, 59] has shown a great impact on video creation [8, 25, 33, 57, 70, 76, 83, 87]. Hence, there is a natural question, *can we build a generic AI system to generate wonderful vlogs automatically?*

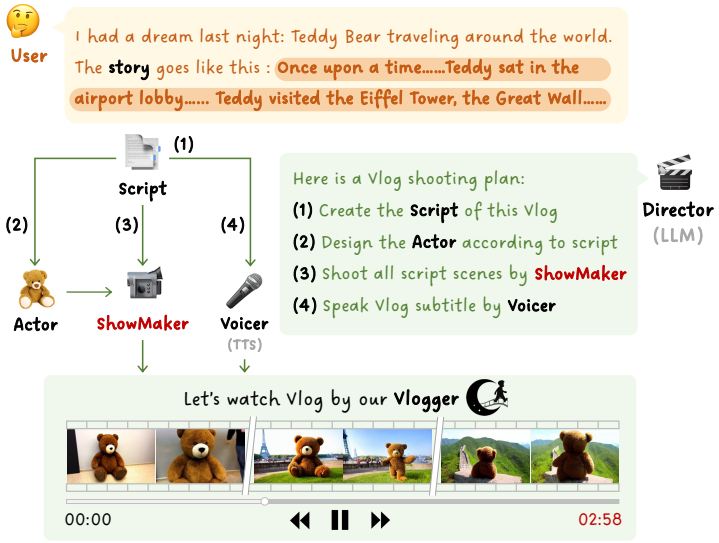

Figure 1. **Overview of Vlogger.** Based on the user story, our Vlogger leverages Large Language Model (LLM) as Director, and decomposes a minute-long vlog generation task into four key stages with Script, Actor, ShowMaker, and Voicer. Moreover, ShowMaker is a novel video diffusion model to generate video snippet of each shooting scene, with script and actor coherence.

Regrettably, the majority of current video diffusion approaches mainly generate short videos with a few seconds, by temporal adaptation of image diffusion models [8, 25, 57, 70, 76, 87]. In contrast, a vlog typically constitutes a *minute*-level long video in the open world. Recently, there have been some attempts in long video generation [67, 83]. However, these early works either require extensive training on large well-captioned long video datasets [83] or suffer from noticeable incoherence of shot changes [67]. Hence, it is still challenging to generate a minute-level

♠ Interns at Shanghai AI Laboratory. ♡ Corresponding authors.

vlog with complex narratives and multiple scenes.

Alternatively, we notice that a successful vlog production is a systematical work in our realistic world, where the key staffs are involved in script creation, actor design, video shooting, and editing [22]. Drawing inspiration from this, we believe that, generating a long-form video blog requires elaborate systematical planning and shooting processes, rather than simply designing a generative model.

Hence, we propose a generic AI system for vlog generation in this paper, namely **Vlogger**, which can smartly address this difficult task by mimicking vlog professionals with various foundation models in the core steps. As shown in Fig. 1, we first hire a Large Language Model (LLM) as *Director* (e.g., GPT-4 [46]), due to its great power of linguistic knowledge understanding. Given a user story, this director schedules a four-step plan for vlog generation. (1) *Script*. First, we introduce a progressive script creation paradigm for the LLM Director. By the coarse-to-fine instructions in this paradigm, the LLM Director can effectively convert a user story into a script, which sufficiently describes the story by a number of shooting scenes and their corresponding shooting duration. (2) *Actor*. After creating the script, the LLM Director reads it again to summarize the actors, and then invokes a character designer (e.g., SD-XL [48]) to generate the reference images of these actors in the vlog. (3) *ShowMaker*. With the guidance of script texts and actor images, we develop a novel ShowMaker as a videographer, which can effectively generate a controllable-duration snippet for each shooting scene, with spatial-temporal coherence. (4) *Voicer*. Finally, the LLM Director invokes a voicer (e.g., Bark [1]) to dub the vlog with script subtitles.

It should be noted that our Vlogger overcomes the challenges previously encountered in long video generation tasks. On one hand, it elegantly decomposes the user story into a number of shooting scenes and designs actor images that can participate in different scenes in the vlog. In this case, it can reduce spatial-temporal incoherence of abrupt shot changes, with explicit guidance of scene texts and actor images. On the other hand, Vlogger crafts individual video snippets for every scene and seamlessly integrates them into a single cohesive vlog. Consequently, this bypasses the tedious training process with large-scale long video datasets. Via such collaboration between top-down planning and bottom-up shooting, Vlogger can effectively transform an open-world story into a minute-long vlog.

Furthermore, we would like to emphasize that, **ShowMaker** is a distinctive video diffusion model designed for generating video snippets of each shooting scene. From the structural perspective, we introduce a novel Spatial-Temporal Enhanced Block (STEB) in this model. This block can adaptively leverage scene descriptions and actor images as textual and visual prompts, which attentively guide ShowMaker to enhance spatial-temporal coherence of script and actors. From the training perspective, we develop a probabilistic mode selection mechanism, which can boost the capacity of ShowMaker by mixed training of Text-to-Video (T2V) generation and prediction. More notably, by sequential combination of generation and prediction mode in the inference stage, ShowMaker can produce a video snippet with a controllable duration. This allows our Vlogger to generate a vlog with a preferable duration, according to the planning of each scene in the script by LLM director.

Finally, our method achieves the state-of-the-art performance on both zero-shot T2V generation and prediction, by expensive experiments within the popular video benchmarks. More importantly, our Vlogger outperforms the well-known long video generation method, *i.e.*, Phenaki [67], despite utilizing only 66.7% of the training videos. Remarkably, our Vlogger is capable of generating over 5-minute vlogs, without loss of script and actor coherence in the video. The models and codes will be released afterward.

2. Related Works

Text-to-Video Generation. Distinct from conventional unconditional and class-conditional video generation [7, 9, 13, 17, 23, 58, 63, 64, 72–74, 81, 85], T2V generation focuses on automatically converting textual descriptions into videos. This is a challenging task, as it involves understanding text semantics and translating it into video content. This often requires powerful cross-modal algorithms [50], large computing resources [15, 29], and extensive video data [3, 55, 56, 75, 76, 80]. Based on the success of diffusion models in the Text-to-Image (T2I) generation [4, 6, 18, 40, 45, 51, 52, 54], a series of such works have been recently transferred to T2V generation [8, 25, 33, 57, 70, 71, 76, 87]. However, whether trained from scratch [33, 57] or finetuned from T2I model [8, 25, 70, 71, 76, 87], most of these approaches mainly work on generating short videos with few seconds from simple descriptions. In contrast, our Vlogger can generate a minute-long vlog with complex stories.

Long Video Generation. The generation of long videos predominantly relies on parallel [83] or autoregressive [67] structures. However, these early works still face challenges for vlog generation. On the one hand, the parallel manner can relax spatial-temporal content incoherence problems by coarse-to-fine generation. However, this approach necessitates extensive and laborious training on large, well-annotated long video datasets [83]. On the other hand, the autoregressive manner can relax heavy data requirements for training long videos, by applying short video generation models iteratively with sliding windows. However, this solution often suffers from noticeable shot change and long-term incoherence [14, 28, 49, 67, 69] which becomes problematic when generating vlogs encompassing complex narratives and multiple scenes. It is worth mentioning that the community has gradually realized that delegating higher-

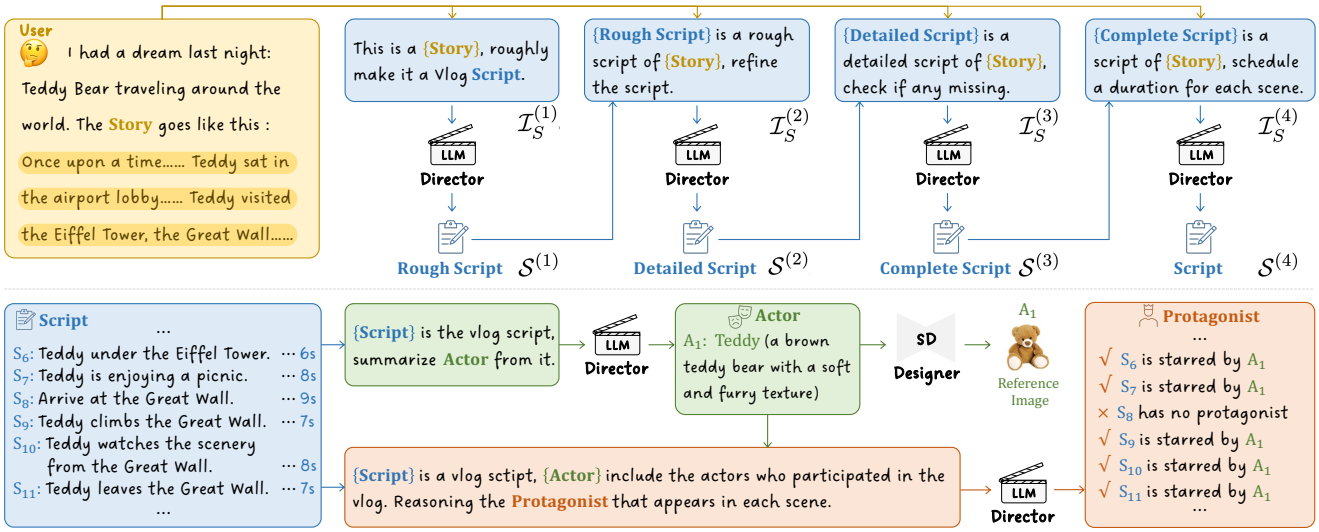

Figure 2. **Top-Down Planning.** Through four rounds of dialogue with the LLM, we gradually convert the user story into the final script. Based on this script, we further extract actor reference images, and then determine which actor would star in each script scene.

order reasoning tasks to LLM is of great help to visual tasks [6, 26, 27, 37, 41, 44, 79, 88, 90]. Our Vlogger introduces LLM to the field of long video generation, and effectively addresses the problems of training burden and content incoherence in the previous methods, by distinct cooperation of top-down planning and bottom-up shooting.

3. Method

In this section, we introduce our Vlogger framework in detail. First, we describe how to make a vlog by planning and shooting with our Vlogger. Then, we further explain the novel design of ShowMaker. As the videographer in our Vlogger, it is critical for video generation in the vlog.

3.1. Overall Framework of Vlogger

To generate a minute-level vlog, our Vlogger leverages LLM as the director, which can effectively decompose this generation task by four key roles within the planning and shooting stages. As shown in Fig. 1, the LLM Director first creates Script and designs Actor in the planning stage. Based on Script and Actor, ShowMaker generates a video snippet for each scene in the shooting stage, and Voicer dubs subtitles of this snippet. Finally, one can combine the snippets of all the scenes as a vlog.

3.1.1 Top-Down Planning

A vlog often comes from a user story that contains diversified content within many shot changes. Clearly, it is challenging to produce a minute-long vlog, by directly feeding such a complex story into video generation models. To this end, we propose to leverage LLM as director and decompose the user story by top-down planning in Fig. 2.

Script Creating. First, we parse the use story into a script, which describes this story explicitly by a number of shooting scenes. In this case, we can generate a video snippet for each shooting scene, instead of learning a long video tediously from the entire story. Since LLM has shown an impressive capacity in language understanding [2, 10, 16, 46, 61, 62, 65, 86], we feed the user story into such a director for script generation. As shown in Fig. 2, we introduce a progressive creation paradigm, which can effectively parse the story by coarse-to-fine steps,

$$\mathcal{S}^{(i)} = \text{LLM}(\mathcal{S}^{(i-1)}, \mathcal{I}_S^{(i)}, \mathcal{U}). \tag{1}$$

Given the user story $\mathcal{U}$ and creation instruction $\mathcal{I}_S^{(i)}$, the LLM Director generates the current script $\mathcal{S}^{(i)}$ from the previous one $\mathcal{S}^{(i-1)}$. More specifically, there are four steps including (1) *Rough*. First, LLM generates a basic draft of the script from the story. (2) *Detailed*. Then, LLM refines the rough script with story details. (3) *Completed*. Next, LLM checks if the detailed script misses the important parts of the story. (4) *Scheduled*. Finally, LLM allocates a shooting duration for each scene in the completed script, according to the scene content. For convenience, we denote the final script as $\mathcal{S}$ in the following. It contains the descriptions of N shooting scenes $\{\mathcal{S}_1, ..., \mathcal{S}_N\}$ and their allocated duration $\{\mathcal{T}_1, ..., \mathcal{T}_N\}$. Additionally, due to the limited pages, please read the full descriptions of instructions and script in the supplementary doc.

Actor Designing. After generating the script, it is time to design actors in the vlog. As shown in Fig. 2, we ask the LLM Director to summarize the actor list $\mathcal{A}$ from script $\mathcal{S}$,

$$\mathcal{A} = \text{LLM}(\mathcal{S}, \mathcal{I}_A), \tag{2}$$

where $\mathcal{I}_A$ is the instruction for actor summarization. Then, according to actor descriptions, the LLM Director invokes

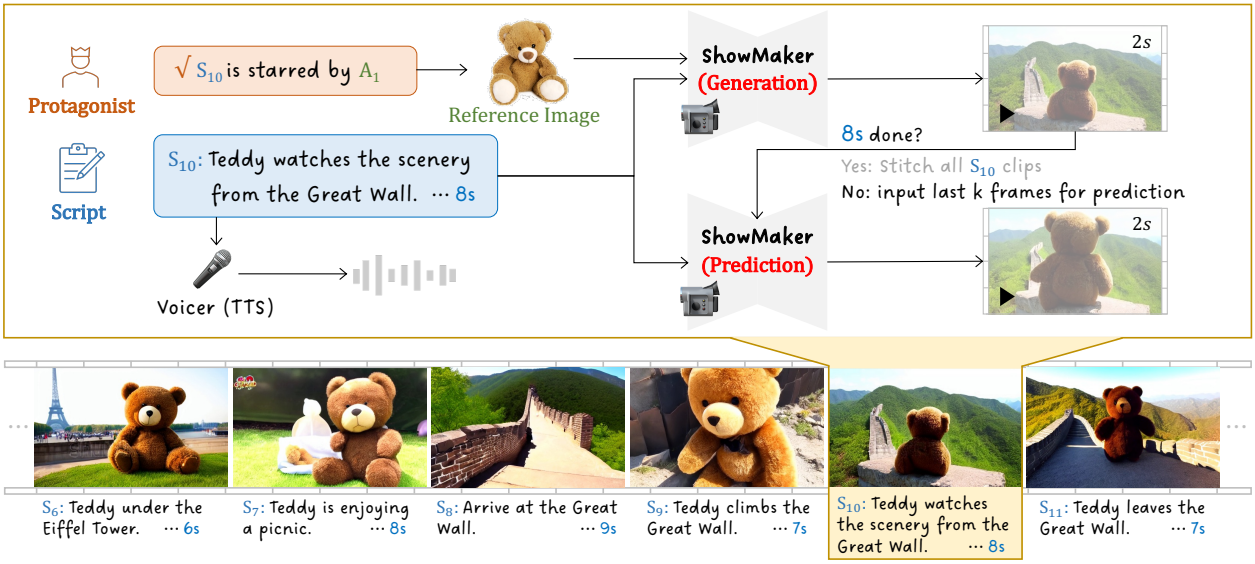

Figure 3. **Bottom-Up Shooting.** For each scene, ShowMaker can generate the video snippet with script and actor coherence, by using script description and actor image as textual and visual prompts. Moreover, ShowMaker can effectively control the snippet duration, by performing generation and prediction sequentially in the inference. Finally, we apply a Text-To-Speech (TTS) model as Voicer for dubbing.

a designer to generate reference images of these actors $\mathcal{R}$,

$$\mathcal{R} = \text{Stable-Diffusion}(\mathcal{A}). \tag{3}$$

We choose Stable Diffusion XL [48] as the character designer, due to its high-quality generation. Finally, based on script $\mathcal{S}$ and actor $\mathcal{A}$, the LLM Director decides the leading actor (*i.e.*, protagonist) in each shooting scene of the script,

$$\mathcal{P} = \text{LLM}(\mathcal{S}, \mathcal{A}, \mathcal{I}_P). \tag{4}$$

where $\mathcal{I}_P$ is the instruction for protagonist selection. The resulting doc $\mathcal{P}$ is aligned with the script $\mathcal{S}$, where $\mathcal{P}_n$ describes which actor appears in the scene $\mathcal{S}_n$. For example, $\{\mathcal{S}_6$ *is starred by* $\mathcal{A}_1\}$ in Fig. 2.

3.1.2 Bottom-Up Shooting

Via the top-down planning above, the LLM Director flexibly decomposes a complex user story into several script scenes and designs actor reference image for each scene. Such a manner largely reduces the difficulty of vlog generation, since we can generate vlogs by bottom-up shooting, *i.e.*, we just need to generate the video snippet for each shooting scene and combine all of them as a vlog.

ShowMaker Shooting. To generate the video snippet of a shooting scene, we introduce a novel ShowMaker as the videographer, which is a video diffusion model with two distinct designs. First, it is important to maintain spatial-temporal coherence of both script and actor in the generated snippet. Hence, our ShowMaker not only takes the scene description $\mathcal{S}_n$ as a textual prompt but also takes actor image of this scene $\mathcal{R}_n$ as a visual prompt. Second, each scene is allocated with a shooting duration in the script. To control the duration of each scene, our ShowMaker contains two learning modes including generation and prediction in Fig.3. Specifically, it starts with the generation mode. For the shooting scene n, we feed its script description $\mathcal{S}_n$ and actor reference image $\mathcal{R}_n$ into ShowMaker,

$$\mathcal{C}_n^{(1)} = \text{ShowMaker}(\mathcal{N}_n^{(1)} \mid \mathcal{S}_n, \mathcal{R}_n, \text{Generate}), \tag{5}$$

which generates the first video clip of this scene $\mathcal{C}_n^{(1)}$ from the noisy clip $\mathcal{N}_n^{(1)}$. If the duration of this clip is smaller than the allocated duration $\mathcal{T}_n$ in the script, we continue to perform the prediction mode, *i.e.*, the last k frames of the current clip $\mathcal{C}_n^{(j)}(k)$ are used as context, when generating the next clip from the noisy input $\mathcal{N}_n^{(j+1)}$,

$$\mathcal{C}_n^{(j+1)} = \text{ShowMaker}(\mathcal{N}_n^{(j+1)} \mid \mathcal{S}_n, \mathcal{C}_n^{(j)}(k), \text{Predict}). \tag{6}$$

Note that, actor reference images are not necessary in the prediction mode, since such an actor's appearance has been shown in the current clip $\mathcal{C}_n^{(j)}(k)$ that is used as input for prediction. This prediction procedure stops until the total duration achieves the allocated $\mathcal{T}_n$ of this scene. Subsequently, we combine all the clips as the video snippet of this scene, *i.e.*, $\mathcal{C}_n = \{\mathcal{C}_n^{(1)}, ..., \mathcal{C}_n^{(J)}\}$.

Voicer Speaking. To enhance the completeness of the vlog, we apply a Text-To-Speech model (e.g., Bark [1]) as a Voicer, which converts the scene description $\mathcal{S}_n$ into the corresponding audio $\mathcal{O}_n = \text{Bark}(\mathcal{S}_n)$. Finally, we add this audio to the corresponding video snippet $\mathcal{V}_n = \mathcal{O}_n \oplus \mathcal{C}_n$, and combine all the sounded video snippets as a complete vlog, *i.e.*, $\mathcal{V} = \{\mathcal{V}_1, ..., \mathcal{V}_N\}$.

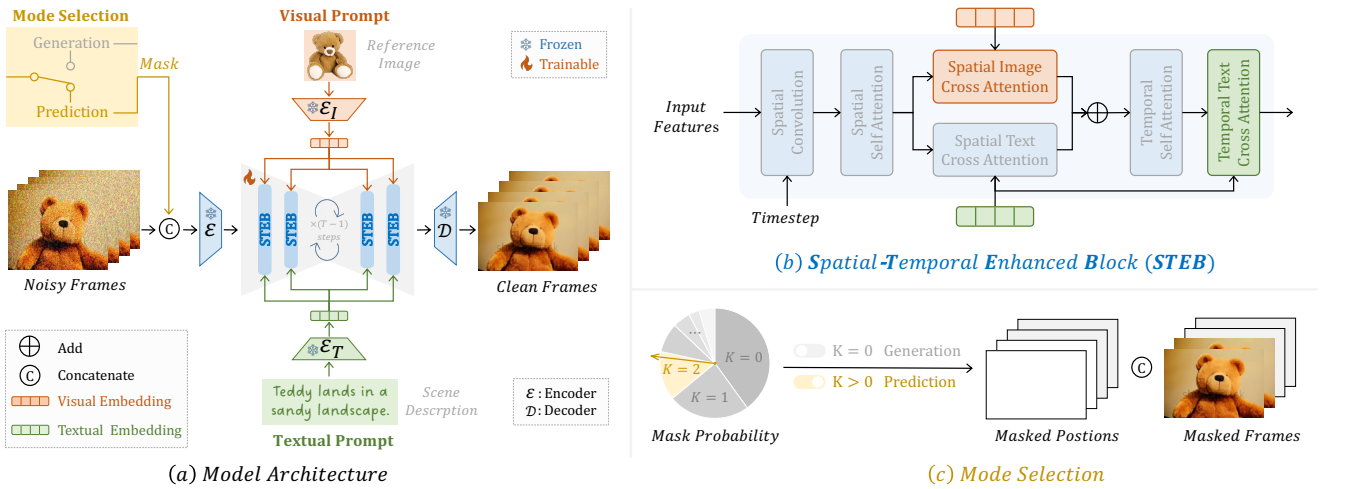

Figure 4. **ShowMaker**. (a) The Overall Architecture. (b) Spatial-Temporal Enhanced Block (STEB). Via spatial-actor and temporal-text cross attention, our STEB can further enhance actor and script coherence in the snippet. (c) Mode Selection. We introduce a mixed training paradigm of T2V generation and prediction, via probabilistic selection of masked frames.

3.2. ShowMaker

As discussed in Section 3.1.2, ShowMaker plays a critical role in generating video snippet of a shooting scene. In this work, we introduce a text-to-video diffusion model for it. It follows the style of latent diffusion model [52]. In the diffusing stage, we add Gaussian noise progressively to the latent code of a training snippet. In the denoising stage, we reconstruct the latent code from the noisy latent code at any iteration step. For simplicity, we just show the denoising stage in Fig.4 (a). First, we forward a noisy training snippet into the encoder of the autoencoder to extract its latent code. Then, we feed this into a denoising U-Net [53] to learn the clean latent code. Finally, we leverage the decoder to reconstruct the original snippet with the clean latent code.

But, compared with the existing video diffusion models [8, 25, 33, 34, 76], our novel ShowMaker contains two distinct designs, in terms of model structure (*i.e.*, Spatial-Temporal Enhanced Block) in Fig.4 (b), and training paradigm (*i.e.*, Mode Selection) in Fig.4 (c).

3.2.1 Spatial-Temporal Enhanced Block (STEB)

To reconstruct the clean latent code of a training video snippet, each block in the denoising U-Net consists of both spatial and temporal operations in previous works [76]. First, spatial operations encode the feature of each frame separately in the snippet. Typically, three operations are inherited from text-to-image generation approaches [4, 45, 51, 52, 54], including spatial ConVolution (CV), spatial Self Attention (SA), and spatial Cross Attention (CA),

$$\mathcal{X}_{cv} = \text{CV-Spatial}(\mathcal{X}_{in}, \mathcal{F}_t), \tag{7}$$

$$\mathcal{X}_{sa} = \text{SA-Spatial}(\mathcal{X}_{cv}), \tag{8}$$

$$\mathcal{X}_{ca} = \text{CA-Spatial-Text}(\mathcal{X}_{sa}, \mathcal{S}_n), \tag{9}$$

where $\mathcal{X}_{in}$ is the noisy feature of the training snippet, and $\mathcal{F}_t$ is the positional embedding of the iteration step t. To guide denoising with the given text, we use the scene description $\mathcal{S}_n$ as Key and Value of cross attention in Eq. (9).

However, we notice that such spatial encoding does not consider actors. Hence, it inevitably suffers from actor incoherence when generating a snippet. To tackle this problem, we introduce a spatial image cross attention,

$$\mathcal{Y}_{ca} = \text{CA-Spatial-Actor}(\mathcal{X}_{sa}, \mathcal{R}_n). \tag{10}$$

For a shooting scene $\mathcal{S}_n$, we leverage the protagonist doc (Eq. 4) to find the corresponding actor in this scene. Then, we leverage actor reference image $\mathcal{R}_n$ as the visual context of spatial cross attention. Subsequently, we enhance spatial embedding with the complementary guidance of both script and actor, *i.e.*, $\mathcal{Z}_{se} = \mathcal{X}_{ca} + \beta\mathcal{Y}_{ca}$ with a weight β.

After spatial encoding, it is time to learn the correlation across frames in the snippet. Hence, the typical operation is to perform self attention along the temporal dimension,

$$\mathcal{Z}_{sa} = \text{SA-Temporal}(\mathcal{Z}_{se}). \tag{11}$$

However, such a temporal operation does not take the constraints of scene text into account. It often leads to text incoherence when generating video snippets. Hence, we introduce a temporal text cross attention,

$$\mathcal{Z}_{ca} = \text{CA-Temporal-Text}(\mathcal{Z}_{sa}, \mathcal{S}_n), \tag{12}$$

where we leverage the scene description $\mathcal{S}_n$ as the textual context of temporal encoding. Via spatial-actor (Eq. 10) and temporal-text (Eq. 12) cross attention, our STEB can further enhance actor and script coherence in the snippet.

Method	FVD (↓)
VideoFactory [71]	410.00
Make-A-Video [57]	367.23
PYoCo [25]	355.19
Ours	**292.43**

(a) Hand-crafted prompt.

Method	FVD (↓)
CogVideo (Chinese) [35]	751.34
CogVideo (English) [35]	701.59
MagicVideo [89]	699.00
Video LDM [8]	550.61
Ours	**525.01**

(b) Class label.

Table 1. **Zero-shot comparison with the state-of-the-art methods on UCF-101**. Hand-crafted prompt from [25].

3.2.2 Mixed Training Paradigm with Mode Selection

As discussed in Section 3.1.2, ShowMaker aims to generate a video snippet with the allocated duration in the script. To this goal, it leverages the combination of both generation and prediction modes in the inference stage. Next, we explain how to train our ShowMaker to learn these modes.

As shown in Fig.4 (c), we design a mode selection mechanism, which selects k frames of the clean snippet as the context of the noisy snippet. The k=0 setting refers to the generation mode since there is no context from the clean snippet. Alternatively, the k>0 setting refers to the prediction mode, since k frames of the clean snippet are already available. The goal is to generate the rest frames of the clean snippet from the noisy one. To integrate both modes into training, we design a probabilistic manner to select k,

$$\mathrm{P}(k) = \begin{cases} \alpha^k - \alpha^{k+1}, & k \in [0, m) \\ \alpha^k, & k = m \end{cases} \tag{13}$$

where $\mathrm{P}(k)$ is the selection probability distribution of k. Moreover, $0<\alpha<1$ and $0\leq m$ are the manual parameters, which respectively control the mode selection tendencies of $\mathrm{P}(k)$ and the maximum number of preserved frames.

After determining k, we introduce a frame mask $\mathcal{M}_k$ on the latent code of the clean snippet $\mathcal{X}_{clean}$,

$$\mathcal{X}_k = \mathcal{X}_{clean} \odot \mathcal{M}_k, \tag{14}$$

where we only preserve k frames and mask the rest of the frames. Then, we concatenate the mask $\mathcal{M}_k$ with the latent code of snippet $\mathcal{X}_{noise}$ and $\mathcal{X}_k$,

$$\mathcal{X}_{in} = \mathrm{Concat}(\mathcal{X}_{noise},\ \mathcal{X}_k,\ \mathcal{M}_k). \tag{15}$$

This produces the input feature $\mathcal{X}_{in}$ for training the denoising U-Net in Section 3.2.1. Via such a concise probabilistic manner, we can effectively integrate both generation (k=0) and prediction (k>0) modes in the training procedure.

4. Experiments

Datasets. To make state-of-the-art comparison, We conduct zero-shot evaluation on the popular video benchmarks, *i.e.*, UCF-101 [60], Kinetics-400 [39], and MSR-VTT [12].

Method	Zero-Shot	Pre-training Videos	FID (↓)
T2V [42]	✗	✗	82.13
SC [5]	✗	✗	33.51
TFGAN [5]	✗	✗	31.76
NUWA [78]	✗	0.97M	28.46
Phenaki [67]	✓	15M	37.74
Ours	✓	10M	**37.23**

Table 2. **Comparison with the state-of-the-art methods on Kinetics-400**. Both under a zero-shot setting, our FID is better than that of Phenaki, with only 66.7% pre-training videos.

Method	Zero-Shot	CLIPSIM (↑)	CLIP-FID (↓)
GODIVA [77]	✗	0.2402	-
NÜWA [78]	✗	0.2439	47.68
CogVideo (Chinese) [35]	✓	0.2614	24.78
CogVideo (English) [35]	✓	0.2631	23.59
Video LDM [8]	✓	0.2929	-
Make-A-Video [57]	✓	0.3049	13.17
PYoCo [25]	✓	-	10.21
Ours	✓	0.2908	**12.67**

Table 3. **Comparison with the state-of-the-art methods on MSR-VTT**. PYoCo [25] sample 59,794 videos for CLIP-FID (noted in gray) while others only sample less than 3,000 videos.

UCF-101 contains videos of 101 action categories. Following [66], we use FVD to evaluate the distance between the generated video and the real video. Kinetics-400 is a dataset comprising videos of 400 action categories. Following [5, 42, 67, 78], we use FID [30] to assess the performance of video generation. MSR-VTT is a video dataset with open-vocabulary captions, wherein CLIPSIM [77] and CLIP-FID [47] are commonly employed for evaluating the T2V generation. Moreover, since these existing benchmarks either have a small number of testing videos or contain only action labels without complex descriptions, we propose to collect an evaluation benchmark for ablation studies. It is called as Vimeo11k, where we collect 11,293 open-world videos along with their captions from 10 mainstream categories in Vimeo. To our best knowledge, it is the largest testing benchmark for zero-shot video generation. More details and experiments can be found in the supplementary doc.

Implementation Details. **(1) Director & Script & Actor & Voicer.** We choose GPT-4 [46] as our LLM director to generate script. The specific instructions can be found in the supplementary doc. Then we employ Stable Diffusion XL [48] and Bark [1] as our designer and voicer to generate reference actor images and convert scripts into speech. **(2) ShowMaker.** We choose SD-1.4 [52] as our base model, and follow [57] to add temporal self attention. Then, we add our temporal text cross attention on top of each temporal self attention. We expand the input channel of the conv-in layer of U-Net from 4 to 9, so that model can take the concatenated feature in Eq. (15) as input. We use zero initialization for newly added channels. We use CLIP ViT-L/14 [50] as text encoder ε_T, VQVAE [20] as autoencoder

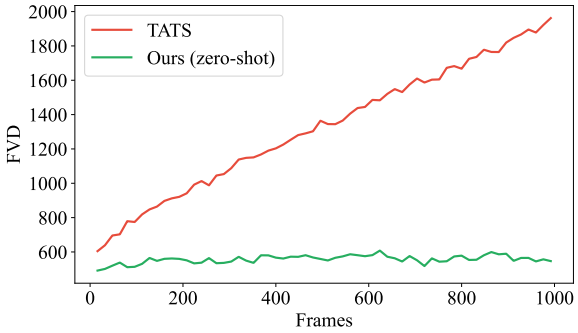

Figure 5. **Long video generation.** Comparison on UCF-101 (1000 frames). The lower FVD, the better generation performance.

Method	CLIP-I (↑)	CLIP-T (↑)	Vlog Duration (↑)
w/o autoregressive [91]	0.5683	0.2752	1min03s
only autoregressive [67]	0.5642	0.2535	3min58s
Ours	**0.6294**	**0.2756**	**3min58s**

Table 4. **Ablation for generation process**. [91] cannot generate a long video in a single scene and both [67, 91] can't refer to images.

consisting of ε and $\mathcal{D}$. Besides, we use OpenCLIP ViT-H/14 [38] as image encoder ε_I, and add spatial image cross attention in the STEB block. The diffusion step T is set to 1000 as [52]. For training, we set the parameter of mode selection as $\alpha = 0.6$ and $m = 6$ in Eq. (13), and β is set to 1. Following [19, 33, 34, 67, 76], we employ publicly available image dataset (*i.e.*, Laion400M [55]) and video dataset (*i.e.*, WebVid10M [3]) for joint training. More training details can be found in the supplementary doc.

4.1. Comparison with state-of-the-art

Tab. 1 shows that, no matter whether the input text is the class label or hand-crafted prompt, our method achieves the best FVD performance of zero-shot video generation on UCF-101. Tab. 2 shows that, compared to Phenaki [67], our method achieves a better FID performance of zero-shot setting on Kinetics-400, but only using 66.7% training videos. Furthermore, our generated videos have a resolution of 320×512, which is higher than Phenaki's 256×256. Tab. 3 shows that, our method achieves remarkably competitive FID and CLIPSIM performance on MSR-VTT.

Furthermore, Fig. 5 illustrates that, we significantly surpass TATS [24] (*i.e.*, the only open-source long video generation model within our knowledge), for generating 1000-frame videos on UCF-101. Moreover, our method does not encounter the issue of TATS, where the video quality continuously declines as the number of frames increases. It is noteworthy that, our method achieves this performance by zero-shot generation, without any finetuning on UCF-101.

4.2. Ablation Study

Vlog Generation Process. Tab. 4 presents a comparison between different generation processes. For a fair comparison, we use the same scripts and employ our ShowMaker as videographer. We compare three approaches including no autoregression at all as MovieFactory [91], fully autoregressive processes in one go as Phenaki [67], and the generation process of our Vlogger. Specifically, We use GPT-4 to generate five stories and go through the Vlogger's planning process to get five vlog scripts. We set 20 different random seeds, so that each of the above generation processes generates 20 different vlogs for each script. We use CLIP-I, CLIP-T [82] and video duration to evaluate the quality of the generated vlogs. The results demonstrate that, even using the same script and videographer, our Vlogger outshines other existing frameworks for preferable vlog generation. More details can be found in the supplementary doc.

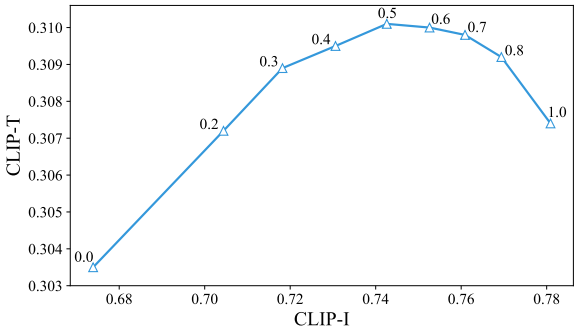

Figure 6. **Ablation for spatial image cross attention.** The values around a point on the curve correspond to the β. While β=0 means the original network without spatial image cross attention.

TTCA	MTP	0/16 (↓)	1/15 (↓)	3/13 (↓)	5/11 (↓)
✗	✗	348.36	217.39	297.37	246.56
✓	✗	333.63	178.27	230.56	193.52
✓	✓	**257.70**	**123.51**	**118.02**	**109.31**

Table 5. **Ablation for temporal text cross attention and mixed training paradigm**. "TTCA" and "MTP" refer to temporal text cross attention and mixed training paradigm respectively. "3/13" denotes predicting the following 13 frames after being given 3 frames of a ground truth video, and we evaluate the FVD between the newly generated 13 frames and the corresponding frames in the ground truth. "0/16", "1/15", "5/11" follow this pattern.

Spatial-Temporal Enhanced Block. Fig. 6 and Tab. 5 evaluate two important operations in our STEB block, *i.e.*, spatial image cross attention and temporal text cross attention. With spatial image cross attention in Fig. 6, the CLIP-I is significantly improved as β increased, while the CLIP-T reached its maximum at β=0.5 on COCO2017 [43]. With temporal text cross attention (TTCA) in Tab. 5, both T2V generation (0/16) and prediction (1/15, 3/13 and 5/11) show a significant improvement on Vimeo11k.

Mixed Training Paradigm. Tab. 5 also presents a model comparison by incorporating our mixed training paradigm or the random mask training method in [11, 14, 21, 36, 68, 84]. The results show that the mixed training paradigm is effective in enhancing the model's T2V generation and prediction capabilities. More details in the supplementary doc.

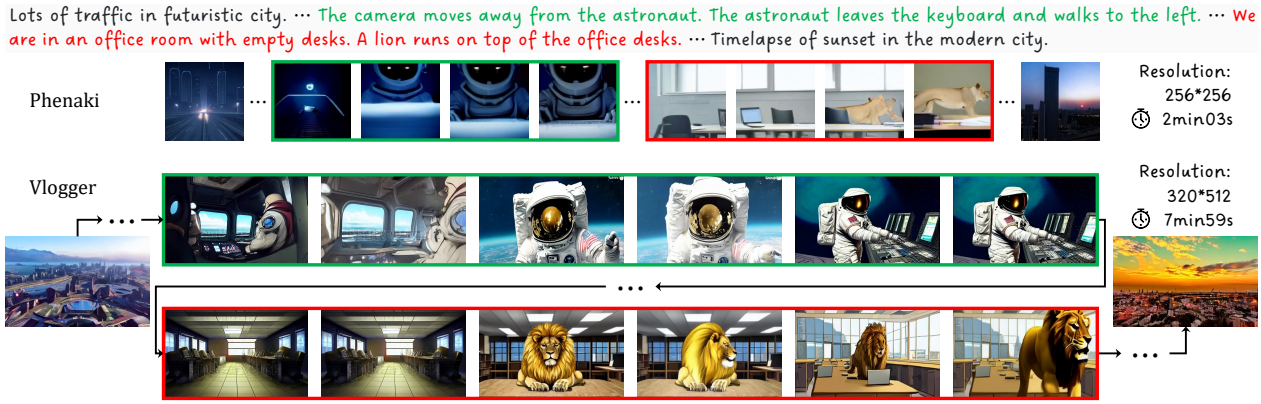

Figure 7. **Qualitative comparison with the state-of-the-art methods on long video generation.** The story and long video of Phenaki [67] are available at phenaki.github.io. The scene diversity and picture quality generated by Vlogger are significantly better than Phenaki.

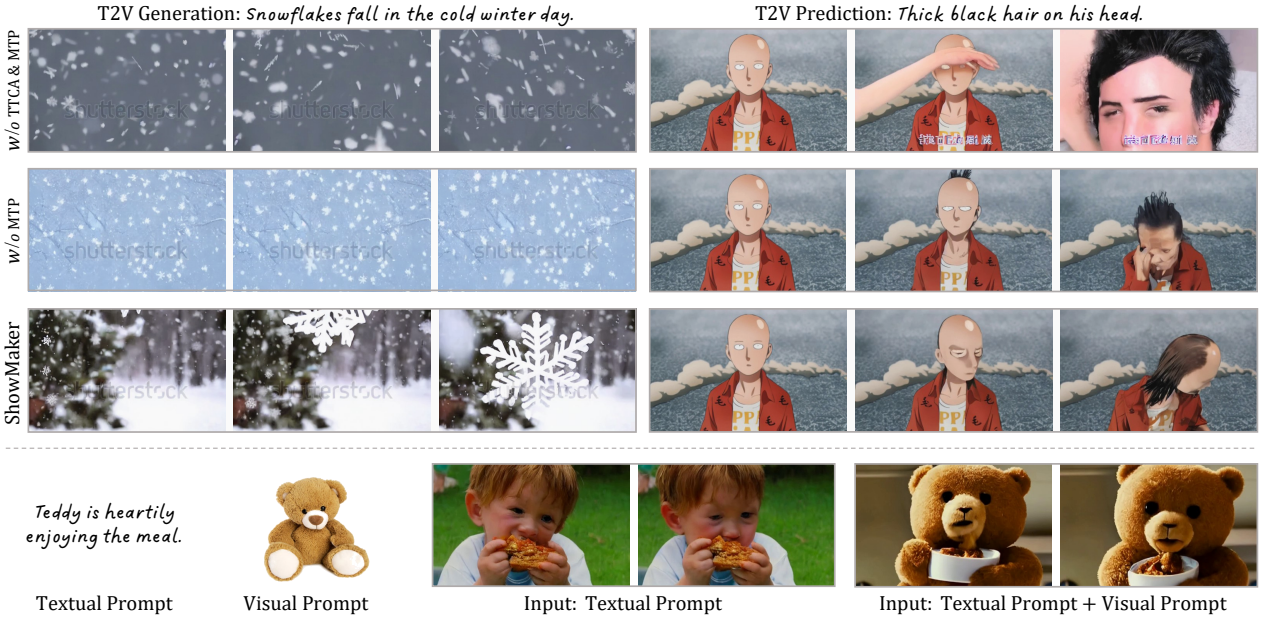

Figure 8. **Qualitative ablation for STEB and training paradigm.** "TTCA" and "MTP" refer to temporal the text cross attention and mixed training paradigm respectively. The sample without a visual prompt corresponds to the model without spatial image cross attention.

4.3. Visualization

First, we visualize long video generation, by comparison with the well-known Phenaki [67]. Note that, since Phenaki does not have the open-sourced codes, we alternatively use the demo shown in its official website. Specifically, we feed the same story description into our Vlogger. As shown in Fig. 7, our Vlogger shows superior and more diverse video content, compared to Phenaki. Moreover, our Vlogger sufficiently exhibits the story with a much longer duration (*i.e.*, 7min59s), according to our LLM-created script.

We further visualize the ablation of T2V video generation and prediction by ShowMaker. As depicted in Fig. 8, ShowMaker had significant improvements in generation and prediction performance, by incorporating our ShowMaker designs. Additionally, it can leverage visual prompts in the spatial actor cross attention, for distinguishing the "Teddy" concept within text prompts.

5. Conclusion

In this paper, we introduce a generic system Vlogger, to generate over 5-minute vlogs from open-world descriptions, without loss of video coherence on script and actor. Moreover, we present a novel video diffusion model ShowMaker for boosting state-of-the-art T2V generation and prediction. Finally, we will release all the models, data and codes afterward, allowing to develop further designs toward long video generation in the open world.

Acknowledgement. This work is supported by the National Key R&D Program of China (NO.2022ZD0160102), the National Natural Science Foundation of China under Grant No. 62102150, and the Science and Technology Commission of Shanghai Municipality under Grant No. 23QD1400800, No. 22511105800.

References

[1] Suno AI. bark. https://github.com/suno-ai/bark#-usage-in-python, 2023. 2, 4, 6

[2] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenhang Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, K. Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Yu Bowen, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xing Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *ArXiv*, abs/2309.16609, 2023. 3

[3] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021. 2, 7

[4] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. *ArXiv*, abs/2211.01324, 2022. 2, 5

[5] Yogesh Balaji1, Martin Renqiang Min, Bing Bai, Rama Chellappa1, and Hans Peter Graf. Conditional gan with discriminative filter generation for text-to-video synthesis. In *IJCAI*, 2019. 6

[6] James Betker, Gabriel Goh, Li Jing, † TimBrooks, Jianfeng Wang, Linjie Li, † LongOuyang, † JuntangZhuang, † JoyceLee, † YufeiGuo, † WesamManassra, † PrafullaDhariwal, † CaseyChu, † YunxinJiao, and Aditya Ramesh. Improving image generation with better captions. 2023. 2, 3

[7] Sarthak Bhagat, Shagun Uppal, Vivian Yin, and Nengli Lim. Disentangling multiple features in video sequences using gaussian processes in variational autoencoders. In *ECCV*, 2020. 2

[8] A. Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, 2023. 1, 2, 5, 6

[9] Tim Brooks, Janne Hellsten, Miika Aittala, Ting-Chun Wang, Timo Aila, Jaakko Lehtinen, Ming-Yu Liu, Alexei A. Efros, and Tero Karras. Generating long videos of dynamic scenes. In *NeurIPS*, 2022. 2

[10] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020. 3

[11] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. In *CVPR*, 2022. 7

[12] Haoran Chen, Jianmin Li, Simone Frintrop, and Xiaolin Hu. The msr-video to text dataset with clean annotations. *Comput. Vis. Image Underst.*, 225:103581, 2021. 6

[13] Xinyuan Chen, Chang Xu, Xiaokang Yang, and Dacheng Tao. Long-term video prediction via criticization and retrospection. *IEEE TIP*, 29:7090–7103, 2020. 2

[14] Xinyuan Chen, Yaohui Wang, Lingjun Zhang, Shaobin Zhuang, Xin Ma, Jiashuo Yu, Yali Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Seine: Short-to-long video diffusion model for generative transition and prediction. *ArXiv*, abs/2310.20700, 2023. 2, 7

[15] François Chollet. On the measure of intelligence. *ArXiv*, abs/1911.01547, 2019. 2

[16] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam M. Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Benton C. Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier García, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Díaz, Orhan Firat, Michele Catasta, Jason Wei, Kathleen S. Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *JMLR*, 24:240:1–240:113, 2022. 3

[17] Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial video generation on complex datasets. *arXiv: Computer Vision and Pattern Recognition*, 2019. 2

[18] Xiaoliang Dai, Ji Hou, Chih-Yao Ma, Sam Tsai, Jialiang Wang, Rui Wang, Peizhao Zhang, Simon Vandenhende, Xiaofang Wang, Abhimanyu Dubey, Matthew Yu, Abhishek Kadian, Filip Radenovic, Dhruv Kumar Mahajan, Kunpeng Li, Yue Zhao, Vladan Petrovic, Mitesh Kumar Singh, Simran Motwani, Yiqian Wen, Yi-Zhe Song, Roshan Sumbaly, Vignesh Ramanathan, Zijian He, Péter Vajda, and Devi Parikh. Emu: Enhancing image generation models using photogenic needles in a haystack. *ArXiv*, abs/2309.15807, 2023. 2

[19] Yatin Dandi, Aniket Das, Soumye Singhal, Vinay P. Namboodiri, and Piyush Rai. Jointly trained image and video generation using residual vectors. *WACV*, pages 3017–3031, 2019. 7

[20] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2020. 6

[21] Tsu-Jui Fu, Licheng Yu, N. Zhang, Cheng-Yang Fu, Jong-Chyi Su, William Yang Wang, and Sean Bell. Tell me what

happened: Unifying text-guided video completion via multimodal masked video generation. In *CVPR*, 2023. 7

[22] Wen Gao, Yonghong Tian, Tiejun Huang, and Qiang Yang. Vlogging: A survey of videoblogging technology on the web. *ACM Computing Surveys*, 2010. 2

[23] Songwei Ge, Thomas Hayes, Harry Yang, Xiaoyue Yin, Guan Pang, David Jacobs, Jia-Bin Huang, and Devi Parikh. Long video generation with time-agnostic vqgan and time-sensitive transformer. In *ECCV*, 2022. 2

[24] Songwei Ge, Thomas Hayes, Harry Yang, Xi Yin, Guan Pang, David Jacobs, Jia-Bin Huang, and Devi Parikh. Long video generation with time-agnostic vqgan and time-sensitive transformer. In *ECCV*, 2022. 7

[25] Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models. *ArXiv*, abs/2305.10474, 2023. 1, 2, 5, 6

[26] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *CVPR*, 2023. 3

[27] Yingqing He, Menghan Xia, Haoxin Chen, Xiaodong Cun, Yuan Gong, Jinbo Xing, Yong Zhang, Xintao Wang, Chao Weng, Ying Shan, and Qifeng Chen. Animate-a-story: Storytelling with retrieval-augmented video generation. *ArXiv*, abs/2307.06940, 2023. 3

[28] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. *ArXiv*, abs/2211.13221, 2023. 2

[29] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Frederick Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *ArXiv*, abs/1712.00409, 2017. 2

[30] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 6

[31] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshops*, 2021. 1

[32] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 1

[33] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey A. Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models. *ArXiv*, abs/2210.02303, 2022. 1, 2, 5, 7

[34] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. *ArXiv*, abs/2204.03458, 2022. 5, 7

[35] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. In *ICLR*, 2023. 6

[36] Tobias Hoppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea Dittadi. Diffusion models for video prediction and infilling. *TMLR*, 2022, 2022. 7

[37] Hanzhuo Huang, Yufan Feng, Cheng Shi, Lan Xu, Jingyi Yu, and Sibei Yang. Free-bloom: Zero-shot text-to-video generator with llm director and ldm animator. *ArXiv*, abs/2309.14494, 2023. 3

[38] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip. https://github.com/mlfoundations/open_clip, 2021. 7

[39] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Apostol Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *ArXiv*, abs/1705.06950, 2017. 6

[40] Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham M. Kakade, Prateek Jain, and Ali Farhadi. Matryoshka representation learning. In *NeurIPS*, 2022. 2

[41] Kunchang Li, Yinan He, Yi Wang, Yizhuo Li, Wen Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *ArXiv*, abs/2305.06355, 2023. 3

[42] Yitong Li, Martin Renqiang Min, Dinghan Shen, David Carlson, and Lawrence Carin. Video generation from text. In *AAAI*, 2017. 6

[43] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 7

[44] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2022. 3

[45] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2021. 2, 5

[46] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023. 2, 3, 6

[47] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *CVPR*, 2022. 6

[48] Dustin Podell, Zion English, Kyle Lacey, A. Blattmann, Tim Dockhorn, Jonas Muller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *ArXiv*, abs/2307.01952, 2023. 2, 4, 6

[49] Haonan Qiu, Menghan Xia, Yong Zhang, Yin-Yin He, Xintao Wang, Ying Shan, and Ziwei Liu. Freenoise: Tuning-free longer video diffusion via noise rescheduling. *ArXiv*, abs/2310.15169, 2023. 2

[50] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2, 6

[51] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125, 2022. 2, 5

[52] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2, 5, 6, 7

[53] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *ArXiv*, abs/1505.04597, 2015. 5

[54] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022. 2, 5

[55] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *ArXiv*, abs/2111.02114, 2021. 2, 7

[56] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022. 2

[57] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. In *ICLR*, 2023. 1, 2, 6

[58] Ivan Skorokhodov, S. Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *CVPR*, 2021. 2

[59] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. 1

[60] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *ArXiv*, abs/1212.0402, 2012. 6

[61] Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Ouyang Xuan, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng Wang. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *ArXiv*, abs/2107.02137, 2021. 3

[62] InternLM Team. Internlm: A multilingual language model with progressively enhanced capabilities. https://github.com/InternLM/InternLM, 2023. 3

[63] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N. Metaxas, and S. Tulyakov. Motion-based generator model: Unsupervised disentanglement of appearance, trackable and intrackable motions in dynamic patterns. In *ICLR*, 2019. 2

[64] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N. Metaxas, and S. Tulyakov. A good image generator is what you need for high-resolution video synthesis. In *ICLR*, 2021. 2

[65] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971, 2023. 3

[66] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. In *ICLR*, 2019. 6

[67] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. In *ICLR*, 2023. 1, 2, 6, 7, 8

[68] Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. Mcvd: Masked conditional video diffusion for prediction, generation, and interpolation. In *NeurIPS*, 2022. 7

[69] Fu Lee Wang, Wenshuo Chen, Guanglu Song, Han-Jia Ye, Yu Liu, and Hongsheng Li. Gen-l-video: Multi-text to long video generation via temporal co-denoising. *ArXiv*, abs/2305.18264, 2023. 2

[70] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *ArXiv*, abs/2308.06571, 2023. 1, 2

[71] Wenjing Wang, Huan Yang, Zixi Tuo, Huiguo He, Junchen Zhu, Jianlong Fu, and Jiaying Liu. Videofactory: Swap attention in spatiotemporal diffusions for text-to-video generation. *ArXiv*, abs/2305.10874, 2023. 2, 6

[72] Yaohui Wang, Piotr Tadeusz Bilinski, François Brémond, and Antitza Dantcheva. G3an: Disentangling appearance and motion for video generation. In *CVPR*, 2020. 2

[73] Yaohui Wang, Piotr Tadeusz Bilinski, François Brémond, and Antitza Dantcheva. Imaginator: Conditional spatio-temporal gan for video generation. *WACV*, pages 1149–1158, 2020.

[74] Yaohui Wang, François Brémond, and Antitza Dantcheva. Inmodegan: Interpretable motion decomposition generative adversarial network for video generation. *ArXiv*, abs/2101.03049, 2021. 2

[75] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, Sen Xing, Guo Chen, Junting Pan, Jiashuo Yu, Yali Wang, Limin Wang, and Yu Qiao. Internvideo: General video foundation models via generative and discriminative learning. *ArXiv*, abs/2212.03191, 2022. 2

[76] Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Pe der Yang, Yuwei Guo, Tianxing Wu, Chenyang Si, Yuming Jiang, Cunjian Chen, Chen Change Loy, Bo Dai, Dahua Lin, Y. Qiao, and Ziwei Liu. Lavie: High-quality video

generation with cascaded latent diffusion models. *ArXiv*, abs/2309.15103, 2023. 1, 2, 5, 7

[77] Chenfei Wu, Lun Huang, Qianxi Zhang, Binyang Li, Lei Ji, Fan Yang, Guillermo Sapiro, and Nan Duan. Godiva: Generating open-domain videos from natural descriptions. *ArXiv*, abs/2104.14806, 2021. 6

[78] Chenfei Wu, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Daxin Jiang, and Nan Duan. Nüwa: Visual synthesis pre-training for neural visual world creation. In *ECCV*, 2022. 6

[79] Chenfei Wu, Sheng-Kai Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models. *ArXiv*, abs/2303.04671, 2023. 3

[80] Hongwei Xue, Tiankai Hang, Yanhong Zeng, Yuchong Sun, Bei Liu, Huan Yang, Jianlong Fu, and Baining Guo. Advancing high-resolution video-language representation with large-scale video transcriptions. In *CVPR*, 2021. 2

[81] Wilson Yan, Yunzhi Zhang, P. Abbeel, and A. Srinivas. Videogpt: Video generation using vq-vae and transformers. *ArXiv*, abs/2104.10157, 2021. 2

[82] Hu Ye, Jun Zhang, Siyi Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *ArXiv*, abs/2308.06721, 2023. 7

[83] Sheng-Siang Yin, Chenfei Wu, Huan Yang, Jianfeng Wang, Xiaodong Wang, Minheng Ni, Zhengyuan Yang, Linjie Li, Shuguang Liu, Fan Yang, Jianlong Fu, Gong Ming, Lijuan Wang, Zicheng Liu, Houqiang Li, and Nan Duan. Nuwa-xl: Diffusion over diffusion for extremely long video generation. In *Annual Meeting of the Association for Computational Linguistics*, 2023. 1, 2

[84] Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G. Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, and Lu Jiang. Magvit: Masked generative video transformer. In *CVPR*, 2023. 7

[85] Sihyun Yu, Jihoon Tack, Sangwoo Mo, Hyunsu Kim, Junho Kim, Jung-Woo Ha, and Jinwoo Shin. Generating videos with dynamics-aware implicit generative adversarial networks. In *ICLR*, 2022. 2

[86] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, P. Zhang, Yuxiao Dong, and Jie Tang. Glm-130b: An open bilingual pre-trained model. *ArXiv*, abs/2210.02414, 2022. 3

[87] David Junhao Zhang, Jay Zhangjie Wu, Jia-Wei Liu, Rui Zhao, Lin Hao Ran, Yuchao Gu, Difei Gao, and Mike Zheng Shou. Show-1: Marrying pixel and latent diffusion models for text-to-video generation. *ArXiv*, abs/2309.15818, 2023. 1, 2

[88] Tianjun Zhang, Yi Zhang, Vibhav Vineet, Neel Joshi, and Xin Wang. Controllable text-to-image generation with gpt-4. *ArXiv*, abs/2305.18583, 2023. 3

[89] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *ArXiv*, abs/2211.11018, 2022. 6

[90] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *ArXiv*, abs/2304.10592, 2023. 3

[91] Junchen Zhu, Huan Yang, Huiguo He, Wenjing Wang, Zixi Tuo, Wen-Huang Cheng, Lianli Gao, Jingkuan Song, and Jianlong Fu. Moviefactory: Automatic movie creation from text using large generative models for language and images. In *ACM MM*, 2023. 7